

**PATENT APPLICATION**

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re the Application of

Kazunori HORIKIRI

Application No.: New U.S. Patent Application

Filed: September 6, 2000

Docket No.: 107196

For: ACCESS PRIVILEGE TRANSFERRING METHOD

#2 / proposed  
pages



**CLAIM FOR PRIORITY**

Director of the U.S. Patent and Trademark Office  
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. JP 11-275702 filed September 29, 1999

In support of this claim, a certified copy of said original foreign application:

  X   is filed herewith.

           was filed on            in Parent Application No.            filed           .

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of this document.

Respectfully submitted,

James A. Oliff  
Registration No. 27,075

Thomas J. Pardini  
Registration No. 30,411

JAO:TJP/cmm/cmm  
Date: September 6, 2000

**OLIFF & BERRIDGE, PLC**  
**P.O. Box 19928**  
**Alexandria, Virginia 22320**  
**Telephone: (703) 836-6400**

**DEPOSIT ACCOUNT USE  
AUTHORIZATION**

Please grant any extension  
necessary for entry;  
Charge any fee due to our  
Deposit Account No. 15-0461

日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出願年月日  
Date of Application:

1999年 9月29日

出願番号  
Application Number:

平成11年特許願第275702号

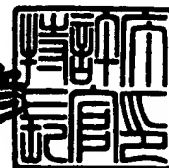
出願人  
Applicant(s):

富士ゼロックス株式会社

2000年 6月 9日

特許庁長官  
Commissioner,  
Patent Office

近藤 隆彦



出証番号 出証特2000-3044721

【書類名】 特許願

【整理番号】 FN99-00280

【提出日】 平成11年 9月29日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/14

【発明の名称】 アクセス権限委譲方法

【請求項の数】 17

【発明者】

    【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0   グリーンテクなかい  
                        富士ゼロックス株式会社内

    【氏名】 堀切 和典

【特許出願人】

    【識別番号】 000005496

    【氏名又は名称】 富士ゼロックス株式会社

    【電話番号】 0462-38-8516

【代理人】

    【識別番号】 100086531

    【弁理士】

    【氏名又は名称】 澤田 俊夫

    【電話番号】 03-5541-7577

【手数料の表示】

    【予納台帳番号】 038818

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書   1

    【物件名】 図面   1

    【物件名】 要約書   1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 アクセス権限委譲方法

【特許請求の範囲】

【請求項 1】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (a) 各クライアントがユーザ情報と秘密情報を保持するステップと、
  - (b) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
  - (c) クライアントが権限情報を生成するステップと、
  - (d) クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、
  - (e) クライアントが、ユーザ情報と権限情報と保護化権限情報を他のクライアントに送信するステップと、
  - (f) 他のクライアントが、ユーザ情報と権限情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
  - (g) サーバが、ステップ (f) において受信した権限情報が有効か否かを検査するステップと、
  - (h) サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、
  - (i) サーバが、ステップ (f) において受信した保護化権限情報とステップ (h) において生成した保護化権限情報とを比較するステップと、
  - (j) ステップ (i) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 2】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上の

クライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するクライアントが、他のクライアントに対してアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (a) サーバと共有すべきユーザ情報と秘密情報を保持するステップと、
- (b) 権限情報を生成するステップと、
- (c) 少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して、他のクライアントに安全に委譲することができる保護化権限情報を生成するステップと、

を具備することを特徴とするアクセス権限委譲方法。

【請求項 3】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するサーバが、アクセス権限を委譲されたクライアントからのアクセス要求に対して安全に応答するためのアクセス権限委譲方法であって、

- (f) ユーザ情報と権限情報と保護化権限情報を含んだアクセス要求を受信するステップと、
  - (g) ステップ (f) において受信した権限情報が有効か否かを検査するステップと、
  - (h) 少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、
  - (i) ステップ (f) において受信した保護化権限情報とステップ (h) において生成した保護化権限情報とを比較するステップと、
  - (j) ステップ (i) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスを許容するステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 4】

前記の所定の演算操作とは、被演算子の各々を連結したビット列に対して一方

向性関数を適用することであることを特徴とする請求項 1、2、又は 3 のいずれかに記載のアクセス権限委譲方法。

【請求項 5】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (A) 各クライアントがユーザ情報と秘密情報を保持するステップと、
- (B) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
- (C) クライアントが権限情報を生成するステップと、
- (D) クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第 1 の保護化権限情報を生成するステップと、
- (E) クライアントが、ユーザ情報と権限情報と第 1 の保護化権限情報を他のクライアントに送信するステップと、
- (F) 他のクライアントが、サーバからチャレンジ文字列を受信するステップと、
- (G) 他のクライアントが、少なくともチャレンジ文字列と第 1 の保護化権限情報とからなる情報に対して所定の演算操作を適用して第 2 の保護化権限情報を生成するステップと、
- (H) 他のクライアントが、ユーザ情報と権限情報と第 2 の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
- (I) サーバが、ステップ (H) において受信した権限情報が有効か否かをチェックするステップと、
- (J) サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第 1 の保護化権限情報を生成するステップと、
- (K) サーバが、少なくともチャレンジ文字列とステップ (J) において生成した第 1 の保護化権限情報とからなる情報に対して所定の演算操作を適用して第 2 の保護化権限情報を生成するステップと、

(L) サーバが、ステップ (H) において受信した第 2 の保護化権限情報とステップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、

(M) ステップ (N) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、  
を具備することを特徴とするアクセス権限委譲方法。

【請求項 6】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、ユーザ情報と権限情報と第 1 の保護化権限情報を譲受したクライアントとサーバとの間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

(F) サーバが、オブジェクトへのアクセスを要求するクライアントに対してチャレンジ文字列を送信するステップと、

(G) クライアントが、少なくともチャレンジ文字列と第 1 の保護化権限情報とからなる情報に対して所定の演算操作を適用して第 2 の保護化権限情報を生成するステップと、

(H) クライアントが、ユーザ情報と権限情報と第 2 の保護化権限情報をサーバに送信して、オブジェクトのアクセスを要求するステップと、

(I) サーバが、ステップ (H) において受信した権限情報が有効か否かを検査するステップと、

(J) サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第 1 の保護化権限情報を生成するステップと、

(K) サーバが、少なくともチャレンジ文字列とステップ (J) において生成した第 1 の保護化権限情報とからなる情報に対して所定の演算操作を適用して第 2 の保護化権限情報を生成するステップと、

(L) サーバが、ステップ (H) において受信した第 2 の保護化権限情報とステップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、

(M) ステップ (N) における比較の結果、両者が一致することに応答して、サーバがオブジェクトに対するアクセスを許容するステップと、

を具備することを特徴とするアクセス権限委譲方法。

【請求項 7】

前記の所定の演算操作とは、被演算子の各々を連結したビット列に対して一方  
向性関数を適用することであることを特徴とする請求項 5 又は 6 のいずれ  
かに記載のアクセス権限委譲方法。

【請求項 8】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上の  
クライアントとがネットワーク接続され、クライアントが所有する権限情報に従  
ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各  
クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するた  
めのアクセス権限委譲方法であって、

- (a) 各クライアントがユーザ情報と秘密情報を保持するステップと、
  - (b) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
  - (c) クライアントが権限情報を生成するステップと、
  - (d) クライアントが、秘密情報を用いて権限情報を暗号化して保護化権限情報  
を生成するステップと、
  - (e) クライアントが、ユーザ情報と保護化権限情報を他のクライアントに送信  
するステップと、
  - (f) 他のクライアントが、ユーザ情報と保護化権限情報をサーバに送信して、  
オブジェクトへのアクセスを要求するステップと、
  - (g) サーバが、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号  
化して、権限情報を生成するステップと、
  - (h) サーバが、ステップ (g) において生成した権限情報が有効か否かを検査  
するステップと、
  - (i) ステップ (h) における有効性の検査結果に従って、オブジェクトに対す  
るアクセスが許容されるステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 9】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上の

クライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するクライアントが、他のクライアントに対してアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (a) サーバと共有すべきユーザ情報と秘密情報を保持するステップと、
  - (b) 権限情報を生成するステップと、
  - (d) 秘密情報を用いて権限情報を暗号化して、他のクライアントに安全に委譲することができる保護化権限情報を生成するステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 1 0】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で動作するサーバが、アクセス権限を委譲されたクライアントからのアクセス要求に対して安全に応答するためのアクセス権限委譲方法であって、

- (f) ユーザ情報と保護化権限情報を含んだアクセス要求を受信するステップと、
  - (g) ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成するステップと、
  - (h) ステップ (g) において生成した権限情報が有効か否かを検査するステップと、
  - (i) ステップ (h) における有効性の検査結果に従って、オブジェクトに対するアクセスを許容するステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 1 1】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するた

めのアクセス権限委譲方法であって、

- (A) 各クライアントがユーザ情報と秘密情報を保持するステップと、
  - (B) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
  - (C) クライアントが権限情報を生成するステップと、
  - (D) クライアントが、秘密情報を用いて権限情報を暗号化して第 1 の保護化権限情報を生成するステップと、
  - (E) クライアントが、ユーザ情報と権限情報と第 1 の保護化権限情報を他のクライアントに送信するステップと、
  - (F) 他のクライアントが、サーバからチャレンジ文字列を受信するステップと、
  - (G) 他のクライアントが、第 1 の保護化権限情報を用いてチャレンジ文字列を暗号化して第 2 の保護化権限情報を生成するステップと、
  - (H) 他のクライアントが、ユーザ情報と権限情報と第 2 の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
  - (I) サーバが、ステップ (H) において受信した権限情報が有効か否かを検査するステップと、
  - (J) サーバが、秘密情報を用いて権限情報を暗号化して第 1 の保護化権限情報を生成するステップと、
  - (K) サーバが、ステップ (J) において生成した第 1 の保護化権限情報を用いてチャレンジ文字列を暗号化して第 2 の保護化権限情報を生成するステップと、
  - (L) サーバが、ステップ (H) において受信した第 2 の保護化権限情報とステップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、
  - (M) ステップ (N) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、
- を具備することを特徴とするアクセス権限委譲方法。

【請求項 12】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、ユ

ーザ情報と権限情報と第 1 の保護化権限情報を譲受したクライアントとサーバとの間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

(F) サーバが、オブジェクトへのアクセスを要求するクライアントに対してチャレンジ文字列を送信するステップと、

(G) クライアントが、第 1 の保護化権限情報を用いてチャレンジ文字列を暗号化して第 2 の保護化権限情報を生成するステップと、

(H) クライアントが、ユーザ情報と権限情報と第 2 の保護化権限情報をサーバに送信して、オブジェクトのアクセスを要求するステップと、

(I) サーバが、ステップ (H) において受信した権限情報が有効か否かを検査するステップと、

(J) サーバが、秘密情報を用いて権限情報を暗号化して第 1 の保護化権限情報を生成するステップと、

(K) サーバが、ステップ (J) において生成した第 1 の保護化権限情報を用いてチャレンジ文字列を暗号化して第 2 の保護化権限情報を生成するステップと、

(L) サーバが、ステップ (H) において受信した第 2 の保護化権限情報とステップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、

(M) ステップ (N) における比較の結果、両者が一致することに応答して、サーバがオブジェクトに対するアクセスを許容するステップと、  
を具備することを特徴とするアクセス権限委譲方法。

### 【請求項 1 3】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに秘密情報を送信するステップと、

第 1 のクライアントが第 2 のクライアントに暗号化鍵を送信するステップと、

第 2 のクライアントが、前記暗号化鍵を用いて前記秘密情報を暗号化した後に、2 次記憶装置に格納するステップと、

を具備することを特徴とする情報管理方法。

【請求項 14】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが、暗号化鍵を用いて秘密情報を暗号化して、保護化秘密情報を生成するステップと、

第 1 のクライアントが第 2 のクライアントに前記保護化秘密情報を送信するステップと、

第 2 のクライアントが前記保護化秘密情報を 2 次記憶装置に格納するステップと、

第 1 のクライアントが、前記暗号化鍵で暗号化された情報を復号化するための復号化鍵を、第 2 のクライアントに送信するステップと、

第 2 のクライアントが、復号化鍵を用いて保護化秘密情報を復号化して、秘密情報を得るステップと、

を具備することを特徴とする情報管理方法。

【請求項 15】

前記暗号化鍵と前記復号化鍵は同一鍵であることを特徴とする請求項 14 に記載の情報管理方法。

【請求項 16】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに秘密情報を送信するステップと、

第 2 のクライアントが、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを保持するステップと、

第 2 のクライアントが復号化鍵を第 1 のクライアントに送信するステップと、

第 2 のクライアントが、前記秘密情報を前記暗号化鍵で暗号化した保護化秘密

情報を 2 次記憶装置に格納するステップと、

第 2 のクライアントが、前記復号化鍵を用いて保護化秘密情報を復号化して前記秘密情報を得るステップと、

を具備することを特徴とする情報管理方法。

【請求項 17】

オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに第 1 の秘密情報を送信するステップと、

第 2 のクライアントが第 1 のクライアントにチャレンジ文字列を送信するステップと、

第 1 のクライアントが前記チャレンジ文字列と第 2 の秘密情報に対して所定の演算処理を適用して暗号鍵を生成するステップと、

第 1 のクライアントが第 2 のクライアントに前記暗号鍵を送信するステップと、

第 2 のクライアントが、前記暗号鍵を用いて前記秘密情報を暗号化して得た保護化秘密情報を 2 次記憶装置に格納するステップと、

を具備することを特徴とする情報管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送するアクセス権限委譲方法に関する。

【0002】

更に詳しくは、本発明は、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジ

ェクトに関するアクセス権限を記述した「ケイパビリティ」をホスト（ユーザ）間で安全に転送するための、アクセス権限委譲方法に関する。

【0003】

【従来の技術】

昨今の情報技術（IT）分野における急速な進歩とともに、ワークステーションやパーソナル・コンピュータなどの各種の汎用コンピュータ・システムが開発・製造され、大学等の研究機関や企業のオフィス、一般家庭内に広く普及している。

【0004】

コンピュータ・システム上では、テキスト形式の文書ファイルや、音声ファイル、画像ファイルなど、デジタイズされた様々な資源オブジェクトを取り扱うことができる。

【0005】

最近では、多くのコンピュータ・システムは、LAN（Local Area Network）やインターネットなどのネットワークに接続され、分散コンピューティング環境に置かれている。分散コンピューティング環境下では、各ユーザはプログラムやデータなど資源オブジェクトの所在を特に認識する必要がなくなる。さらに、コンピュータにおいて実行される手続きやメソッドも、ネットワーク上で分散して保持され、管理されている。

【0006】

例えば、ネットワーク上のある1つのコンピュータ上で動作しているプロセスが、他のコンピュータ上で動作するプロセスの手続きを呼び出して実行させるという、「遠隔手続き呼び出し（RPC：Remote Procedure Call）」、若しくは、「遠隔メソッド呼び出し（RMI：Remote Method Invocation）」と呼ばれる手法も広汎に採り入れられている。実行される手続きのインターフェースを予め記述し、呼び出し側及び実行側の双方のコンピュータに配置しておくことによって、このような遠隔手続き呼び出しを好適に実現することができる。

【0007】

その反面、ファイルなどの夫々の共有資源オブジェクト (Object) に対して、誰 (Subject) にどの程度のアクセス操作 (Verb) を許容するか、といった「アクセス権 (Access Privileges)」に関する管理方法や管理システムが重要な技術的課題となってくる。アクセス権の制御は、S (Subject: 主体) - V (Verb: アクセス操作) - O (Object: 操作対象としてのファイルなど) の関係に見立てて記述し、管理することができる。Sすなわちサブジェクトとは、ファイル操作を望むユーザ、ユーザ・アカウント、又はユーザ識別情報とほぼ同義と把握されたい。

## 【0008】

ここで言うオブジェクト (O) に対するアクセス操作 (V) の権限としては、例えば以下のようなものが挙げられる。

- (1) ファイルを読み出す権利
- (2) ファイルに書き込む (更新する) 権利
- (3) ファイルを新規作成する権利
- (4) ファイルを削除する権利
- (5) ファイルを検索する権利
- (6) ファイル名などのファイル属性を変更する権利

## 【0009】

全てのユーザに対して無制限なアクセス権を許容してしまうと、ファイルが勝手に複製、改竄、又は削除されてしまうなど悪用や乱用が横行する危険が発生する。この結果、あるユーザは予期しない不利益を被り、時として致命傷を負う。アクセス権を適切に管理しなければ、分散コンピューティング環境の崩壊をも招来するであろう。

## 【0010】

オブジェクトに対するアクセス権の管理方法として、「アクセス制御マトリックス」なる概念が、当業界において古くから知られている。このアクセス制御マトリックスは、以下の【表1】に示すように、誰 (Subject) に、ファイルなどの各オブジェクト (Object) に対してどの程度のアクセス権 (Verb) を与えたかを一覧表示したテーブル形式で記述される。

【0011】

【表1】

Alexが持つ権限情報 (capability)

	#FILE1	#FILE2	#FILE3	.....
Alex	Read/Write	Read		
Bob	Read	Read/Write		
Cod				
⋮	⋮	⋮	⋮	⋮

#FILE についてのアクセス  
制御リスト (ACL)

【0012】

【表1】に示す例では、例えば”Alex”なる主体すなわちユーザ・アカウントには、オブジェクトとしての”File #1”に対する読み出し (Read) 及び書き込み (Write) というアクセス操作が許容されているが、”File #2”に対しては読み出しのみ許され…、”Bob”なる主体には…となっている。

【0013】

アクセス制御マトリックスを各カラム毎に切り出したものは、オブジェクトが各サブジェクトすなわちユーザに対して許容した操作権限を記述した情報であり、「アクセス制御リスト (Access Control List: ACL)」と呼ばれる。例えば、【表1】に示した例では、”File #1”に相当するカラムは、該オブジェクトに対してAlex, Bob, Codの各々に与えられた操作権限を示したアクセス制御リストである。サーバ用若しくは開発プラットフォーム用のオペレーティング・システム (OS) として広汎に普及している”Unix”は、簡略形式のアクセス制御リストを使用している。

## 【0014】

また、アクセス制御マトリックスを各行毎に切り出したものは、サブジェクトすなわちユーザが各々のオブジェクトに対して許容された操作権限を記述した情報であり、「権限情報」若しくは「ケイパビリティ (Capability)」と呼ばれる。例えば、「表1」に示すアクセス制御マトリックスの第1行はユーザ”Alex”に関するケイパビリティに相当し、また、第2行はユーザ”Bob”に関するケイパビリティに相当する。

## 【0015】

ケイパビリティは、オブジェクトへのアクセスを許容する一種の鍵の役割を持つ。すなわち、ケイパビリティを所有するサブジェクトすなわちユーザには、オブジェクトへのアクセスが許可される。例えば、無数のホストがTCP/IP (Transmission Control Protocol/Internet Protocol) 接続されてなるネットワーク (例えばインターネット) 上では、ケイパビリティをURL (Uniform Resource Locator) 文字列の中で記述して、HTTP (Hypertext Transfer Protocol) メッセージとしてホスト間で交換することもできる。

## 【0016】

例えば、AlexがCodにケイパビリティを渡すことによって、Codは、各オブジェクトに対してAlexと同等の操作権限を所有することができる。ケイパビリティを引き継いだCodは、Alexの不在中も、Alexの代理として各オブジェクトにアクセスして、Alexの任務を代行することができる。

## 【0017】

しかしながら、ケイパビリティを無制限に譲渡しあるいは貸し与えてしまうと、ケイパビリティの悪用・乱用により、思わぬ不利益を受ける可能性がある。このため、ケイパビリティの譲渡者は、ケイパビリティに有効期限や使用許可回数を付加したり、オブジェクトの操作権限を制限する (例えばフル・アクセスを読み出しのみに限定する) など、権限内容を弱めてケイパビリティを渡すことが好ましい。

【0018】

また、ケイパビリティを譲り受けたサブジェクトが、ケイパビリティの内容を簡単に解釈できるような形式で渡したのでは、譲受者はケイパビリティの内容を改竄して権限を勝手に強めたり、無断でケイパビリティを複製して、オブジェクトを無制限にアクセス操作してしまうであろう。また、ケイパビリティの転送を傍受する第三者も、同様に、オブジェクトを不正に操作できてしまう。

【0019】

以上を要言するならば、ネットワーク上の各ホスト間でケイパビリティを安全に転送することがオブジェクトを保護する上で重要な技術的課題となる訳である。

【0020】

オブジェクトのアクセス制御に関しては、既に幾つかの技術が提案されている。

【0021】

例えば特開平5-81204号公報には、分散型コンピュータ・システムにおけるアクセス制御について開示されている。同公報によれば、ケイパビリティに相当する特権属性証明書(PAC)の代理使用を制御すると同時に、PACを多くの目的のために使用できる方法が提供されている。すなわち、PACを配布する際には、これに開始者実体に相当する開始者資格付与属性を含めるようにするとともに、暗号的に開始者資格付与属性を有した暗号キーを開始者実体に配布するようになっている。

【0022】

しかしながら、特開平5-81204号公報に開示された方法によれば、使用期限やアクセス回数など、ケイパビリティのバリエーションには対応できない。また、開始実体者がケイパビリティを弱めたPACを自由に作成する方法については一切言及していない。

【0023】

また、特開平9-319659号公報には、非分散型のコンピュータ・システムにおいて、各ユーザに異なるケイパビリティを割り当てる方式について開示さ

れている。同公報によれば、コンピュータ機能の全体はケイパビリティ・セットを有するイベント・セットに細分化されており、ユーザがコンピュータ・システム上で実行しようとする特定のジョブに応じてケイパビリティが与えられるようになっている。

【0024】

しかしながら、特開平9-319659号公報に係る発明は、ケイパビリティを安全に守ることができない分散環境に適用されたものではない。また、同公報は、オブジェクト間でケイパビリティの検査を安全に行う方法については一切言及していない。

【0025】

また、特開平9-251425号公報には、分散システムにおけるシステム資源へのアクセスのセキュリティ制御について開示している。同公報に開示されるセキュリティ制御システムによれば、グループ識別標識を記憶しておき、これをターゲット・オブジェクトに結び付け、次にメンバーシップ検査を使用して、ターゲット・オブジェクトへのアクセスを要求するクライアントがターゲットに対するアクセス権限を有するグループ・メンバーであるか否かを判断するようになっている。

【0026】

しかしながら、特開平9-251425号公報では、使用期限やアクセス回数など、ケイパビリティのバリエーションには対応する手法については開示していない。また、あるケイパビリティの保有者が新たにケイパビリティを生成したりこれを無効化する手法についても一切言及していない。

【0027】

また、分散OS（オペレーティング・システム）として当業界では周知の”Amoeba”は、譲渡可能で且つ偽造が困難なケイパビリティを提供している。これは以下の機構により実現される。

- (1) クライアントがオブジェクト生成要求をサーバに送信する。
- (2) サーバは、オブジェクトを生成して、オブジェクト番号と乱数を割り当てる。乱数は、オブジェクト番号でインデックス付けされたオブジェクト・テーブル

ルに格納される。

(3) サーバは、乱数をキーとして権限と乱数を暗号化したチェック・フィールドを含むケイパビリティを生成する。

(4) ケイパビリティをクライアントに返す。

(5) クライアントが、サーバに対して、ケイパビリティを提示したアクセス要求を出す。

(6) サーバは、ケイパビリティからオブジェクト番号とチェック・フィールドを抽出し、オブジェクト・テーブルに格納されている乱数を基にチェック・フィールドから権限と乱数を復号化する。

(7) 復号の結果得られた乱数がオブジェクト・テーブルに格納された乱数と一致するかどうか検証する。

(8) 要求された操作がケイパビリティに記述された権限に合致すれば、サーバはその操作を実行する。

#### 【0028】

Amoebaでは、さらに、ケイパビリティの保持者が権限を弱めた新しいケイパビリティを生成するために、以下の機構も備えている。

(1) クライアントとサーバがN個の可換な一方向性関数を共有する。

(2) クライアントが第2の権限を生成する。

(3) クライアントが第1の権限と第2の権限との差分の権限に該当する番号と対応付けた一方向性関数を全て適用して、第2のチェック・フィールドを生成する。

(4) サーバが、第2の権限と第2のチェック・フィールドを含むケイパビリティを受信する。

(5) サーバは、権限フィールドに従い、一方向性関数を順次適用して、クライアントが提示したチェック・フィールドと一致した場合に、その操作を実行する。

#### 【0029】

しかしながら、Amoebaは、N個の可換な一方向性関数を用いてN個の権限の可否を記述する方法については規定しているが、使用権限や回数といった権

限についての多くのバリエーションに対応する点については考慮していない。また、ケイパビリティの無効化に関しては、サーバで保持されている乱数を変更することによってあるオブジェクトについての全てのケイパビリティを無効化する手法が規定されているが、特定のケイパビリティを無効化する手法については言及していない。例えば、あるケイパビリティの保持者が生成したケイパビリティや、これから派生したケイパビリティのみを無効化する点は一切規定していない。また、Amoebaではオブジェクトにアクセスするためのオブジェクト番号自体については一切プロテクトがかけられていない。

【0030】

また、Web上で公開されたBjorn N. Freeman-Bensonの論文”Using the Web to Private Information—or—A Short Paper About Password Protection Without Client Modification”（URLは”http://www1.cern.ch/www94/PrelimProcs.html”）では、機密性のあるURL（すなわちケイパビリティ）の取り扱いについて開示されている。同論文に記載された方式は以下の手順に従う。

- （１）サーバがログイン名とパスワードの組からなるパスワード・リストを生成・保持する。また、サーバは暗号鍵を保持している。
- （２）クライアントがログイン名とパスワードの組からなるメッセージをサーバに送信する。
- （３）サーバは、パスワード・リストを検索し、クライアントから送信されたログイン名とパスワードの組が見つかれば、このログイン名とパスワードを暗号鍵で暗号化した文字列をアクセス・キーとして、URLに含めてクライアントに送信する。
- （４）クライアントは、受け取ったURLを用いてサーバにアクセス要求する。
- （５）サーバは、受信したURLからアクセス・キーを抽出し、暗号鍵を用いて復号化して、もとのログイン名とパスワードの組を得る。そして、これがパスワード・リストに登録されているかどうかを検査する。

(6) 登録されていれば、サーバは対応するオブジェクトへのアクセスを許可する。

【0031】

また、同論文では、クライアントがパスワードの変更を行うことでURLを無効化することができる点を言及している。

【0032】

しかしながら、同論文では、ある1つのオブジェクトについての正当で且つ異なるURL（すなわちケイパビリティ）を複数生成する点については開示していない。

【0033】

以上を総括すると、従来技術では下記のような問題があった。すなわち、

(1) 使用期限や有効使用回数などの権限内容に多種多様なバリエーションに対応しながらケイパビリティを安全に管理し委譲することが困難である。

(2) 該当するオブジェクトに関する全てのケイパビリティを無効化するのではなく、一部のケイパビリティのみを無効化することは困難である。

(3) ケイパビリティを所有するサブジェクトが、ケイパビリティが持つ権限の内容を変更した（弱めた）ケイパビリティを自由に生成し、且つ、これを安全に管理し委譲し、あるいは無効化することができない。

【0034】

【発明が解決しようとする課題】

本発明の目的は、サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することにある。

【0035】

本発明の更なる目的は、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」を、ホスト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供すること

にある。

【0036】

本発明の更なる目的は、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、他のサブジェクトに安全に委譲することができる、優れたアクセス権限委譲方法を提供することにある。

【0037】

本発明の更なる目的は、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、生成されたケイパビリティをオブジェクトの管理者が安全に検査することができる、優れたアクセス権限委譲方法を提供することにある。

【0038】

【課題を解決するための手段及び作用】

本発明は、上記課題を参酌してなされたものであり、その第1の側面は、オブジェクトを提供する1以上のサーバと、オブジェクトを要求する1以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるコンピューティング環境下で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (a) 各クライアントがユーザ情報と秘密情報を保持するステップと、
- (b) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
- (c) クライアントが権限情報を生成するステップと、
- (d) クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、
- (e) クライアントが、ユーザ情報と権限情報と保護化権限情報を他のクライアントに送信するステップと、
- (f) 他のクライアントが、ユーザ情報と権限情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
- (g) サーバが、ステップ(f)において受信した権限情報が有効か否かを検査するステップと、

(h) サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して保護化権限情報を生成するステップと、

(i) サーバが、ステップ (f) において受信した保護化権限情報とステップ (h) において生成した保護化権限情報とを比較するステップと、

(j) ステップ (i) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、

を具備することを特徴とするアクセス権限委譲方法である。

【0039】

また、本発明の第2の側面は、オブジェクトを提供する1以上のサーバと、オブジェクトを要求する1以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

(A) 各クライアントがユーザ情報と秘密情報を保持するステップと、

(B) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、

(C) クライアントが権限情報を生成するステップと、

(D) クライアントが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第1の保護化権限情報を生成するステップと、

(E) クライアントが、ユーザ情報と権限情報と第1の保護化権限情報を他のクライアントに送信するステップと、

(F) 他のクライアントが、サーバからチャレンジ文字列を受信するステップと、

(G) 他のクライアントが、少なくともチャレンジ文字列と第1の保護化権限情報とからなる情報に対して所定の演算操作を適用して第2の保護化権限情報を生成するステップと、

(H) 他のクライアントが、ユーザ情報と権限情報と第2の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、

(I) サーバが、ステップ (H) において受信した権限情報が有効か否かを検査するステップと、

(J) サーバが、少なくとも権限情報と秘密情報とからなる情報に対して所定の演算操作を適用して第 1 の保護化権限情報を生成するステップと、

(K) サーバが、少なくともチャレンジ文字列とステップ (J) において生成した第 1 の保護化権限情報とからなる情報に対して所定の演算操作を適用して第 2 の保護化権限情報を生成するステップと、

(L) サーバが、ステップ (H) において受信した第 2 の保護化権限情報とステップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、

(M) ステップ (N) における比較の結果、両者が一致することに応答してオブジェクトに対するアクセスが許容されるステップと、  
を具備することを特徴とするアクセス権限委譲方法である。

【 0 0 4 0 】

また、本発明の第 3 の側面は、オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (a) 各クライアントがユーザ情報と秘密情報を保持するステップと、
- (b) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
- (c) クライアントが権限情報を生成するステップと、
- (d) クライアントが、秘密情報を用いて権限情報を暗号化して保護化権限情報を生成するステップと、
- (e) クライアントが、ユーザ情報と保護化権限情報を他のクライアントに送信するステップと、
- (f) 他のクライアントが、ユーザ情報と保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
- (g) サーバが、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成するステップと、
- (h) サーバが、ステップ (g) において生成した権限情報が有効か否かを検査するステップと、

(i) ステップ (h) における有効性の検査結果に従って、オブジェクトに対するアクセスが許容されるステップと、  
を具備することを特徴とするアクセス権限委譲方法である。

【0041】

また、本発明の第4の側面は、オブジェクトを提供する1以上のサーバと、オブジェクトを要求する1以上のクライアントとがネットワーク接続され、クライアントが所有する権限情報に従ったオブジェクトに対するアクセス操作が許容されるオブジェクト空間上で、各クライアント間及びクライアントーサーバ間でアクセス権限を安全に委譲するためのアクセス権限委譲方法であって、

- (A) 各クライアントがユーザ情報と秘密情報を保持するステップと、
- (B) サーバが各クライアントのユーザ情報と秘密情報を保持するステップと、
- (C) クライアントが権限情報を生成するステップと、
- (D) クライアントが、秘密情報を用いて権限情報を暗号化して第1の保護化権限情報を生成するステップと、
- (E) クライアントが、ユーザ情報と権限情報と第1の保護化権限情報を他のクライアントに送信するステップと、
- (F) 他のクライアントが、サーバからチャレンジ文字列を受信するステップと、
- (G) 他のクライアントが、第1の保護化権限情報を用いてチャレンジ文字列を暗号化して第2の保護化権限情報を生成するステップと、
- (H) 他のクライアントが、ユーザ情報と権限情報と第2の保護化権限情報をサーバに送信して、オブジェクトへのアクセスを要求するステップと、
- (I) サーバが、ステップ (H) において受信した権限情報が有効か否かを検査するステップと、
- (J) サーバが、秘密情報を用いて権限情報を暗号化して第1の保護化権限情報を生成するステップと、
- (K) サーバが、ステップ (J) において生成した第1の保護化権限情報を用いてチャレンジ文字列を暗号化して第2の保護化権限情報を生成するステップと、
- (L) サーバが、ステップ (H) において受信した第2の保護化権限情報とステ

ップ (K) において生成した第 2 の保護化権限情報とを比較するステップと、  
 (M) ステップ (N) における比較の結果、両者が一致することに対応してオブジェクトに対するアクセスが許容されるステップと、  
 を具備することを特徴とするアクセス権限委譲方法である。

【0042】

また、本発明の第 5 の側面は、オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに秘密情報を送信するステップと、  
 第 1 のクライアントが第 2 のクライアントに暗号化鍵を送信するステップと、  
 第 2 のクライアントが、前記暗号化鍵を用いて前記秘密情報を暗号化した後に、2 次記憶装置に格納するステップと、  
 を具備することを特徴とする情報管理方法である。

【0043】

また、本発明の第 6 の側面は、オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが、暗号化鍵を用いて秘密情報を暗号化して、保護化秘密情報を生成するステップと、

第 1 のクライアントが第 2 のクライアントに前記保護化秘密情報を送信するステップと、

第 2 のクライアントが前記保護化秘密情報を 2 次記憶装置に格納するステップと、

第 1 のクライアントが、前記暗号化鍵で暗号化された情報を復号化するための復号化鍵を、第 2 のクライアントに送信するステップと、

第 2 のクライアントが、復号化鍵を用いて保護化秘密情報を復号化して、秘密情報を得るステップと、

を具備することを特徴とする情報管理方法である。

【 0 0 4 4 】

本発明の第 6 の側面に係る情報管理方法において、前記暗号化鍵と前記復号化鍵は、同一鍵すなわち「対称鍵暗号方式」の鍵であっても、あるいは、暗号化鍵（秘密鍵）で暗号化した情報は復号化鍵（公開鍵）でしか復号化できないような「公開鍵暗号方式」における鍵の組み合わせであってもよい。

【 0 0 4 5 】

また、本発明の第 7 の側面は、オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに秘密情報を送信するステップと、

第 2 のクライアントが、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを保持するステップと、

第 2 のクライアントが復号化鍵を第 1 のクライアントに送信するステップと、

第 2 のクライアントが、前記秘密情報を前記暗号化鍵で暗号化した保護化秘密情報を 2 次記憶装置に格納するステップと、

第 2 のクライアントが、前記復号化鍵を用いて保護化秘密情報を復号化して前記秘密情報を得るステップと、

を具備することを特徴とする情報管理方法である。

【 0 0 4 6 】

また、本発明の第 8 の側面は、オブジェクトを提供する 1 以上のサーバと、オブジェクトを要求する 1 以上のクライアントとがネットワーク接続されたオブジェクト空間上で、各クライアント間及び／又はクライアントーサーバ間で秘密情報を安全に管理するための情報管理方法であって、

第 1 のクライアントが第 2 のクライアントに第 1 の秘密情報を送信するステップと、

第 2 のクライアントが第 1 のクライアントにチャレンジ文字列を送信するステップと、

第1のクライアントが前記チャレンジ文字列と第2の秘密情報に対して所定の演算処理を適用して暗号鍵を生成するステップと、

第1のクライアントが第2のクライアントに前記暗号鍵を送信するステップと、

第2のクライアントが、前記暗号鍵を用いて前記秘密情報を暗号化して得た保護化秘密情報を2次記憶装置に格納するステップと、

を具備することを特徴とする情報管理方法である。

【0047】

【作用】

本発明は、例えばLAN (Local Area Network) やインターネットのような分散コンピューティング環境において適用される。かかる分散コンピューティング環境下では、オブジェクトを提供する1以上のサーバと、オブジェクトを要求する1以上のクライアントとがネットワーク接続され、いわゆる「オブジェクト空間」が形成されている。オブジェクト空間の一例は、各ホストがTCP/IP (Transmission Control Protocol/Internet Protocol) 接続された分散コンピューティング環境下で展開される、HTML (Hyper Text Markup Language) 形式で記述されたHTTP (Hyper Text Transfer Protocol) オブジェクトが提供されるWWW (World Wide Web) 情報空間である。

【0048】

オブジェクト空間上では、各クライアントのオブジェクトに対するアクセス権限は「権限情報」若しくは「ケイパビリティ (capability)」という形式で記述される。したがって、クライアントは、基本的には、自身のユーザ情報と自身に付与された権限情報を、オブジェクトを管理し提供するサーバに提示することによって、オブジェクトへのアクセスが許容される。

【0049】

さらに、クライアントから権限情報を引き継いだ他のクライアントも、元のクライアントのユーザ情報と権限情報とを提示することによって、オブジェクトへ

のアクセスが許容される。しかしながら、無制限に権限情報を譲渡してしまうと、譲受者による権限情報の乱用や不十分な安全管理のために、譲渡者であるクライアントは、権限情報の悪用や乱用、あるいは権限情報の無断の複製や改竄によって、不測の不利益を被りかねない。

## 【0050】

そこで、権限情報の譲渡者は、オブジェクトに対する無制限のアクセスを排除するために、権限内容を弱めた権限情報（例えば、有効期限や使用許可回数などを付加したり、オブジェクトの操作権限を制限した権限情報）を委譲することが好ましい。また、委譲した後に、権限情報が勝手に改竄されないように（例えば、勝手に有効期限を書き換えたり、操作権限を強化したりされないように）、権限情報を安全に送信できるように対策を講じる必要がある。

## 【0051】

本発明は、各クライアントとサーバ間で、クライアントのユーザ情報と秘密情報が共有されていることを前提とする。秘密情報は、パスワード等を指し、例えば所定の情報に暗号化関数を適用する際の秘密鍵として利用することができる。

## 【0052】

本発明の第1の側面によれば、権限情報を委譲するクライアントは、自身を持つ権限内容を弱めた権限情報を生成し、生成した権限情報と秘密情報とを連結したビット列に対して所定の演算操作を適用することで、改竄などの悪用の危険を排した保護化権限情報を生成することができる。

## 【0053】

ここで言う所定の演算操作の一例は「一方向性関数」である。一方向性関数は、逆関数を求めることが極めて困難な関数であり、該関数を適用する前の引数の値を推測することを不可能にする作用がある。したがって、秘密情報を知らない第三者が保護化権限情報を勝手に改竄することはできない。

## 【0054】

クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、保護化権限情報とを送信する。そして、他のクライアントは、譲受したユーザ情報、権限情報、及び保護化権限情

報を添付して、サーバに対してオブジェクトに対するアクセス要求を行うことができる。

【0055】

一方のサーバは、まず、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

【0056】

次いで、サーバは、アクセス要求メッセージで受け取ったユーザ情報と秘密情報とを連結したビット列に対して、所定の演算操作、例えば一方向性関数を適用して、保護化権限情報を再現してみる。そして、この再現された保護化権限情報を、アクセス要求メッセージで受け取った保護化権限情報と比較照合することで、アクセス要求クライアントが正当な権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

【0057】

また、本発明の第2の側面に係るアクセス権限委譲方法では、認証処理のために、「チャレンジ文字列」と呼ぶ1回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を採用する。

【0058】

まず、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成する。そして、生成した権限情報と秘密情報とを連結したビット列に対して一方向性関数などの演算操作を適用することで、悪用の危険を排した第1の保護化権限情報を生成する。秘密情報を知らない第3者は、第1の保護化権限情報を勝手に改竄することはできない。

【0059】

クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、第1の保護化権限情報を送信する

## 【0060】

サーバは、クライアントからのアクセス要求に対して、チャレンジ文字列を返信する。これに対し、クライアントは、チャレンジ文字列と第1の保護化権限情報を連結したビット列に対して一方向性関数などの所定の演算操作を適用して、悪用の危険を排した第2の保護化権限情報を生成する。チャレンジ文字列を知らない第3者は、第2の保護化権限情報を勝手に改竄することはできない。

## 【0061】

次いで、クライアントは、譲受したユーザ情報、権限情報、及び第2の保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を改めて行う。

## 【0062】

サーバは、アクセス要求に応答して、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かをチェックする。チェック結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

## 【0063】

次いで、サーバは、アクセス要求メッセージで受け取ったユーザ情報と秘密情報とを連結したビット列に対して一方向性関数などの所定の演算操作を適用して、第1の保護化権限情報を再現してみる。さらに、チャレンジ文字列と第1の保護化権限情報を連結したビット列に対して一方向性関数などの所定の演算操作を適用して、第2の保護化権限情報を再現してみる。そして、この再現された第2の保護化権限情報を、アクセス要求メッセージで受け取った第2の保護化権限情報と比較照合することで、アクセス要求クライアントが正当な権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全にチェックすることができる訳である。

## 【0064】

また、本発明の第3の側面に係るアクセス権限委譲方法は、委譲する権限情報

を保護化する所定の演算操作として、一方向性関数ではなく暗号化関数を用いたものである。クライアントとサーバの間で共有される秘密情報を暗号化関数の暗号鍵として用いることができる。暗号化方式としては、対称秘密鍵方式又は公開鍵暗号方式<sup>/\*</sup>の双方を適用することができる。

## 【0065】

まず、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成し、次いで、秘密情報を用いて権限情報を暗号化して、悪用の危険を排した保護化権限情報を生成する。したがって、秘密情報を知らない第3者は、保護化権限情報を勝手に改竄することはできない。

## 【0066】

クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と保護化権限情報を送信する。そして、他のクライアントは、譲受したユーザ情報と保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を行う。

## 【0067】

一方のサーバは、まず、ユーザ情報に対応する秘密情報を用いて保護化権限情報を復号化して、権限情報を生成する。

## 【0068】

次いで、サーバは、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かを検査する。検査結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

## 【0069】

また、本発明の第4の側面に係るアクセス権限委譲方法は、第3の側面に対していわゆる「チャレンジ文字列」と呼ぶ1回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を適用したものである。「チャレンジ・レスポンス認証」は「ゼロ知識証明」の一種である（前述）。

## 【0070】

まず、権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成する。そして、秘密情報を用いて権限情報を暗号化して、悪用の危険を排した第1の保護化権限情報を生成する。秘密情報を知らない第3者は、第1の保護化権限情報を勝手に改竄することはできない。

## 【0071】

クライアントは、アクセス権限を委譲したい他のクライアントに対して、自身のユーザ情報と、新たに生成した権限情報と、第1の保護化権限情報を送信する。

## 【0072】

サーバは、クライアントからのアクセス要求に対して、チャレンジ文字列を返信する。これに対し、クライアントは、第1の保護化権限情報を用いてチャレンジ文字列を暗号化して、悪用の危険を排した第2の保護化権限情報を生成する。チャレンジ文字列を知らない第3者は、第2の保護化権限情報を勝手に改竄することはできない。

## 【0073】

次いで、クライアントは、譲受したユーザ情報、権限情報、及び第2の保護化権限情報を添付して、サーバに対してオブジェクトに対するアクセス要求を改めて行う。

## 【0074】

サーバは、アクセス要求に応答して、権限情報が有効か否か、すなわち、オブジェクトに対してクライアントが要求する操作が権限情報で許容された範囲内か否かをチェックする。チェック結果が否定的であれば、該アクセス要求は不正なアクセスとして拒絶される。

## 【0075】

次いで、サーバは、秘密情報を用いて権限情報を暗号化して、第1の保護化権限情報を再現してみる。さらに、再現された第1の保護化権限情報を用いてチャレンジ文字列を暗号化して、第2の保護化権限情報を再現してみる。そして、この再現された第2の保護化権限情報を、アクセス要求メッセージで受け取った第2の保護化権限情報と比較照合することで、アクセス要求クライアントが正当な

権限を有するか否かを安全に判断することができる。両者が一致すればアクセス要求は受理され、両者が一致しなければアクセス要求は不正アクセスとして拒絶される。すなわち、サーバは、委譲されたアクセス権限を安全に検査することができる訳である。

## 【 0 0 7 6 】

上述したように、本発明の第 1 乃至第 4 の側面に係るアクセス権限委譲方法によれば、クライアントは、悪用の危険を排した保護化権限情報を送信することで他のクライアントに対してアクセス権限を安全に委譲することができる。しかしながら、アクセス権限を譲受した他のクライアントが、保護化権限情報をハード・ディスクのような 2 次記憶装置に無防備に格納してしまうと、第三者がハード・ディスクを攻撃したり不正侵入したりして、保護化権限情報自体が漏洩してしまう危険がある。漏洩した結果、第三者による保護化権限情報を用いた不正アクセスを回避することは困難となってしまう。

## 【 0 0 7 7 】

本発明の第 5 乃至第 8 の側面に係る情報管理方法は、このような技術的課題を鑑みたものであり、保護化権限情報のような秘密性の高い情報を 2 次記憶装置に安全に格納する方法を提供するものである。

## 【 0 0 7 8 】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

## 【 0 0 7 9 】

## 《注釈》

\*：対称秘密鍵方式とは、通信の相手どうしで同一の秘密鍵を共有する方式であり、暗号化するために用いた秘密鍵と同じ鍵を用いて暗号化情報を復号化することができる。これに対し、公開鍵暗号方式とは、一方の鍵で暗号化した暗号化情報は他方の鍵でしか復号化できないという性質を持つ 2 個の鍵の組み合わせで暗号化する方式であり、一方の鍵をユーザ個人が秘密裏に保持する「秘密鍵」として保持し、他方の鍵を第 3 者に公開する「公開鍵」として用いるのが一般的である。例えば公開鍵を用いて暗号化することによって秘密鍵の所有者に秘密文書を

安全に送信することができる。また、秘密鍵を用いて暗号化した署名を送信することにより、受信者は公開鍵を用いて署名を認証することができる。

【0080】

【発明の実施の形態】

本発明は、例えばLAN (Local Area Network) やインターネットのような分散コンピューティング環境において適用される。ネットワーク上の各コンピュータ・システムすなわち「ホスト」は、TCP/IP (Transmission Control Protocol/Internet Protocol) のような所定の通信プロトコルに従って相互接続されている。

【0081】

かかる分散コンピューティング環境下では、オブジェクトを提供する1以上のサービス・オブジェクトと、オブジェクトを要求する1以上のクライアントとがネットワーク接続され、いわゆる「オブジェクト空間」が形成されている。例えば、インターネット上で公開されている広域情報提供システムWWW (World Wide Web) では、HTML (Hyper Text Markup Language) 形式で記述されたハイパーテキスト・オブジェクトが、HTTP (Hyper Text Transfer Protocol) プロトコルに従って提供される。

【0082】

各オブジェクトは、URL (Uniform Resource Locator) と呼ばれる所定書式の文字列によってオブジェクト空間上で一意に指定される。URLとは、ネットワークなどの資源空間上で資源の名称やその在り処を特定するための文字列であり、スキーム名 (プロトコル名) : // ホスト名 (ドメイン名) : ポート番号 / パス名 (ファイル名) という形式で記述される (周知)。クライアントは、所望のHTTPオブジェクトに関するURLを含んだHTTPメッセージを送信することによって、HTTPサーバに対してオブジェクトへのアクセス (すなわちオブジェクトに対する操作) を要求することができる。

【0083】

このようなオブジェクト空間上では、各クライアントのHTTPオブジェクトに対するアクセス権限は「権限情報」若しくは「ケイパビリティ (c a p a b i l i t y)」という形式で記述される。各クライアントは、基本的には、自身のユーザ情報と自身に付与された権限情報をサーバすなわちサービス・オブジェクトに提示することによって、オブジェクトへのアクセスが許容される。HTTPプロトコルでは、ケイパビリティをURL文字列の中で記述して、HTTPメッセージとしてホスト間で交換することも許容される。

## 【0084】

また、クライアントは、権限情報を含んだURLを他のクライアントに引き渡すという形式で、オブジェクトに対するアクセス権限を委譲することができる。但し、既に「従来の技術」の欄でも説明したように、無制限に権限情報を譲渡してしまうと、譲渡者であるクライアントは、権限情報の悪用や乱用、権限情報の無断の複製や改竄によって、不測の不利益を被りかねない。

## 【0085】

以下で説明するように、本発明の各実施形態では、権限情報の譲渡者は、オブジェクトに対する不正なアクセスを排除するために、権限内容を弱めた権限情報（例えば、有効期限や使用許可回数などを付加したり、オブジェクトの操作権限を弱めた権限情報）を委譲することにした。また、委譲した後に、権限情報が勝手に改竄されないように（例えば、勝手に有効期限を書き換えたり、操作権限を強化したりされないように）、権限情報を安全に送信できるように対策を講じた。

## 【0086】

## 1. 第1の実施形態

図1には、本発明の第1の実施形態に係る分散コンピューティング環境を模式的に示している。第1の実施形態は、権限情報に一方方向性関数MDを適用することによってアクセス権限の安全な委譲を実現するものである。以下、各部について説明する。

## 【0087】

インターネットのようなネットワーク10上には、無数のホストがTCP/I

P接続されている。ホストの一部は、HTTPオブジェクトを提供するHTTPサーバ300…であり、また、他の一部は、サーバに対してオブジェクトへのアクセスを要求するクライアント100, 200…である。なお、サーバの実体は、HTTPサーバ・アプリケーションを稼動する汎用コンピュータであり、また、クライアントは、WWWブラウザなどのクライアント・アプリケーションを稼動する汎用コンピュータである。

## 【0088】

図1に示す例では、HTTPサーバ300は、“http://www300”なるURLで表されるものとする。また、HTTPサーバ300は、アクセス制御オブジェクト301と、参照番号391～39Nで示されるN個のHTTPオブジェクトを所有している。各HTTPオブジェクト391～39Nはそれぞれ“http://www300/object391”, …, “http://www300/object39N”なるURLで示されるものとする。また、アクセス制御オブジェクト301は、各HTTPオブジェクト391～39Nに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト301は、HTTPオブジェクト391～39Nと同じサーバ300上に存在する必要は必ずしもなく、例えば別のホスト（図示しない）上に存在してHTTPサーバ300から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

## 【0089】

各クライアント100…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。図1に示す例では、クライアント100は、ユーザ情報“userid1”とパスワード“password1”を保持しているものとする。

## 【0090】

HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…に関するユーザ情報とパスワードの組み合わせ(userid1, password1), (userid2, password2)…を「パスワード管理表」（仮称）として保管している。

【0091】

本実施形態では、HTTP1.1のBasic Authentication Scheme（基本認証法）に従ってHTTPメッセージが交換されるものとする。すなわち、クライアント100は、以下に示すようなHTTPリクエストを発行することにより、HTTPオブジェクト391のGETメソッドを起動することができる。

【0092】

【数1】

```
GET /object391 HTTP/1.1
Authorization: Basic authdata
```

【0093】

但し、“authdata”は、ユーザ情報とパスワードを連結したビット列userid1:password1を、改行のないbase64エンコード法でエンコードした文字列である。なお、ここで言う“base64”とは、MIME (Multipurpose Internet Mail Extensions) を利用した電子メールの添付ファイルの符号化方式であり、文字列や添付ファイルを全て64個のアルファベットや数字などのASCII (American Standard Code for Information Interchange) コードに変換する。

【0094】

次いで、クライアント100がクライアント200に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【0095】

権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでもよい）。

【0096】

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時

● 有効期間終了日時

【0097】

ここで、クライアント100が、クライアント200にアクセス権限を委譲するために、以下の権限情報文字列 `capability1` を生成したとする。

【0098】

【数2】

`((/object391), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)`

【0099】

上記の権限情報文字列は、” `/object391` ” で表されるHTTPオブジェクト391に対して、GETメソッドのみを、世界時で1999年の4月24日10時0分48秒から1999年の4月25日10時0分48秒の間だけ実行することのできる権利を表している。

【0100】

本実施例では、クライアント200は、HTTPサーバ300のみを対象とするため、オブジェクト名としてサーバ内の名前のみを取り扱う。但し、一般には、オブジェクト名としてURL等のグローバルな識別子を用いることにより、複数のHTTPサーバのオブジェクトを識別するようにしてもよい。

【0101】

次いで、クライアント100は、この権限情報 `capability1` に対して秘密情報としての `password1` を文字列後方からビット連結して、一方方向性関数MD (Message Digest) を適用して、以下に示す保護化権限情報 `capabilityMD1` を生成する。

【0102】

【数3】

`MD(((/object), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)+password1)`

【0103】

一方向性関数MDは、逆関数を求めることが極めて困難な関数であり、該関数MDを適用する前の引数の値を推測することを不可能にする作用がある。したがって、秘密情報password1を知らない第3者が保護化権限情報capabilityMD1を勝手に改竄することはできない。

【0104】

次いで、クライアント100は、少なくとも、

- userid1 (ユーザ情報)
- capability1 (権限情報)
- capabilityMD1 (保護化権限情報)

からなる情報を、他のクライアント200に対して送信する。

【0105】

既に述べたように、秘密情報password1を知らない第3者（クライアント200を含む）は、保護化権限情報capabilityMD1を改竄することができない。したがって、クライアント100は、他のクライアント200に対して、HTTPオブジェクト391に対するアクセス権限を安全に委譲することができる訳である。

【0106】

他方、上記の情報を受信したクライアント200は、クライアント100の代理として、HTTPオブジェクト391にアクセスするサービスを提供する。

【0107】

クライアント200は、HTTPオブジェクト391にアクセスするためには、少なくとも以下の情報を含む要求メッセージをHTTPサーバ300に送信する。

【0108】

- オブジェクト名
- メソッド名
- ユーザ名
- 権限情報

- 権限情報のメッセージ・ダイジェスト

【0109】

本実施例では、以下に示すような要求メッセージを送信する。

【0110】

【数4】

```
GET /object391 HTTP/1.1
Authorization: Capability cap1
```

【0111】

ここで、文字列cap1は、ユーザ名と権限情報と権限情報のメッセージ・ダイジェストを” : ”を用いて連結して、改行のないbase64エンコード法（前述）でエンコードした文字列であるとする。

【0112】

HTTPサーバ300のアクセス制御オブジェクト301は、上記の要求メッセージ中のAuthorizationフィールドから、以下の情報を取り出す。

- ユーザ情報
- 権限情報
- 権限情報のメッセージ・ダイジェスト

【0113】

さらに、アクセス制御オブジェクト301は、取り出したこれらの情報を基に、以下に示すアルゴリズム1（図2を参照のこと）を実行することによって、クライアント200がHTTPオブジェクト391に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【0114】

アルゴリズム1：

（ステップ1）Authorizationフィールドから抽出したユーザ名をuseridとする。

【0115】

（ステップ2）Authorizationフィールドから抽出した権限情報を

capabilityとする。

【0116】

(ステップ3) Authorizationフィールドから抽出した権限情報のメッセージ・ダイジェストをcapabilityMDとする。

【0117】

(ステップ4) useridに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをpasswordとする。他方、対応するパスワードが登録されていない場合は、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【0118】

(ステップ5) 権限情報capabilityと秘密情報passwordとをビット連結して、一方向性関数MDを適用し、その結果をdigestとする。

【0119】

(ステップ6) digestを、要求メッセージに含まれるcapcapabilityMDと比較する。両者が等しくない場合には、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【0120】

(ステップ7) Authorizationフィールドから取り出されたcapabilityを基に、クライアント200が要求するメソッド(すなわちオブジェクトに対する処理内容)、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がcapabilityで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【0121】

アルゴリズム1を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト301は、HTTPオブジェクト391に対してGETメソッドを起動するメッセージを送信する。

【0122】

図3には、本実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【0123】

トランザクションの前提として、クライアント100は、HTTPサーバ300に対するアクセスのアカウント（利用者資格）として、ユーザ情報”userid1”とパスワード”password1”を保持しているものとする。また、HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…についてのユーザ情報とパスワードの組み合わせ（userid1, password1）,（userid2, password2）…を「パスワード管理表」として保管している。

【0124】

クライアント100は、HTTPオブジェクト391へのアクセスに関し、クライアント200に与えてもよい権限情報capability1を作成する（Tr1）。

【0125】

次いで、クライアント100は、権限情報capability1の後ろに自己の秘密情報password1をビット連結したビット列に一方方向性関数MDを適用して、保護化権限情報capabilityMD1を作成する（Tr2）。

【0126】

次いで、クライアント100は、ユーザ情報userid1と権限情報capability1、及び保護化権限情報capabilityMD1をクライアント200に送信することで、アクセス権限を安全に委譲する（Tr3）。

【0127】

アクセス権限を委譲されたクライアント200は、HTTPサーバ300に対してHTTPオブジェクト391へのアクセス要求を送信する（Tr4）。このアクセス要求には、クライアント100のユーザ名userid1、権限情報capability1、及び保護化権限情報capabilityMD1を添付

する。

#### 【0128】

HTTPサーバ300のアクセス制御オブジェクト301は、クライアント200のアクセス要求の正当性を検証する(Tr5)。該検証は、上述のアルゴリズム1(図2を参照のこと)が規定する処理手順に従う。

#### 【0129】

検証結果が成功裡に終了した場合には、アクセス制御オブジェクト301はアクセス要求を受理する(Tr6)。より具体的には、要求されているメソッド(例えばGETメソッド)を起動する。

#### 【0130】

### 2. 第2の実施形態

次いで、本発明の第2の実施形態について説明する。第2の実施形態は、上記と同様、図1に示すような分散コンピューティング環境に適用可能である。第2の実施形態は、権限情報に一方向性関数を適用するとともに、「チャレンジ文字列」と呼ぶ1回限り使用する数字を基に暗号化してやり取りする「チャレンジ・レスポンス認証」を採用するものである。

#### 【0131】

本実施形態において、HTTPサーバ300は、"http://www300"なるURLで表されるものとする。また、HTTPサーバ300は、アクセス制御オブジェクト301と、参照番号391~39Nで示されるN個のHTTPオブジェクトを所有している。各HTTPオブジェクト391~39Nはそれぞれ"http://www300/object391", ..., "http://www300/object39N"なるURLで示されるものとする。

#### 【0132】

アクセス制御オブジェクト301は、各HTTPオブジェクト391~39Nに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト301は、HTTPオブジェクト391~39Nと同じサーバ300上に存在する必要は必ずしもなく、例えば別のサーバ(図示しない)上に存

在してHTTPサーバ300から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【0133】

各クライアント100…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント100は、ユーザ情報”userid1”とパスワード”password1”を保持しているものとする。

【0134】

HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…に関するユーザ情報とパスワードの組み合わせ(userid1, password1), (userid2, password2)…を「パスワード管理表」(仮称)として保管している。

【0135】

本実施形態では、HTTP1.1のDigest Authentication Scheme (基本認証法)に従ってHTTPメッセージが交換されるものとする。すなわち、クライアント100は、以下に示すようなHTTPリクエストを発行することにより、HTTPオブジェクト391のGETメソッドを起動することができる。

【0136】

【数5】

```
GET /object391 HTTP/1.1
Authorization: Digest digest-response
```

【0137】

但し、”digest-response”は、HTTP1.1の規定に基づき生成された認証データである。

【0138】

次いで、クライアント100がクライアント200に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【0139】

権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでもよい）。

【0140】

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時
- 有効期間終了日時

【0141】

ここで、クライアント100が、クライアント200にアクセス権限を委譲するために、以下の権限情報文字列 `capability1` を生成したとする。

【0142】

【数6】

`((/object391), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)`

【0143】

上記の権限情報文字列は、“`/object391`” で表されるHTTPオブジェクト391に対して、GETメソッドのみを、世界時で1999年の4月24日10時0分48秒から1999年の4月25日10時0分48秒の間だけ実行することのできる権利を表している。

【0144】

本実施例では、クライアント200は、HTTPサーバ300のみを対象とするため、オブジェクト名としてサーバ内での名前のみを取り扱う。但し、一般には、オブジェクト名としてURL等のグローバルな識別子を用いることにより、複数のHTTPサーバのオブジェクトを識別するようにしてもよい。

【0145】

次いで、クライアント100は、この権限情報 `capability1` に対して秘密情報としての `password1` を文字列後方からビット連結して、一方方向性関数MD (Message Digest) を適用して、以下に示す保護化

権限情報 `capabilityMD1` を生成する。

【0146】

【数7】

`MD((/object), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)+password1)`

【0147】

一方向性関数MDは、逆関数を求めることが極めて困難な関数であり、該関数MDを適用する前の引数の値を推測することを不可能にする作用がある。したがって、秘密情報 `password1` を知らない第3者が保護化権限情報 `capabilityMD1` を勝手に改竄することはできない。

【0148】

次いで、クライアント100は、少なくとも、

- `userid1` (ユーザ情報)
- `capability1` (権限情報)
- `capabilityMD1` (保護化権限情報)

からなる情報を、他のクライアント200に対して送信する。

【0149】

既に述べたように、秘密情報 `password1` を知らない第3者（クライアント200を含む）は、保護化権限情報 `capabilityMD1` を改竄することができない。したがって、クライアント100は、他のクライアント200に対して、HTTPオブジェクト391に対するアクセス権限を安全に委譲することができる訳である。

【0150】

他方、上記の情報を受信したクライアント200は、クライアント100の代理として、HTTPオブジェクト391にアクセスするサービスを提供する。

【0151】

クライアント200は、HTTPオブジェクト391にアクセスするために、まず、以下に示すようなメッセージをHTTPサーバ300に送信する。

【0152】

【数8】

GET /object391 HTTP1.1

【0153】

この場合、要求メッセージに認証データが含まれていないため、HTTPサーバ300は、以下のような応答メッセージをクライアント200に送信する。

【0154】

【数9】

HTTP/1.1 401 Unauthorized

WWW-Authenticate: Capability

realm="private@fujixerox.co.jp",

qop="auth, auth-int",

nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",

opaque="a9f0171e7f40e415aa951069a403ecaf"

【0155】

上記のうち、nonceに代入された文字列は、「チャレンジ・レスポンス認証」を行うための「チャレンジ文字列」である。「チャレンジ・レスポンス認証」とは、パスワード認証を安全に実施するための技術であり、1回限り使用するチャレンジ文字列を基に暗号化してメッセージ交換する方式である。

【0156】

クライアント200は、少なくともチャレンジ文字列nonceと保護化権限情報capabilityMD1とをビット連結した文字列のメッセージ・ダイジェストを計算する。この場合の例で言えば、以下の式を計算することになる。  
すなわち、

【0157】

【数10】

MD("0a093dad98c11d0f600cfc7102dd2f0e8c"+capabilityMD1)

【0158】

また、その他の文字列もさらに連結するようにしてもよい。例えば、HTTP  
/1.1のように以下の式を計算してもよい。

【0159】

【数11】

```
MD(MD(userid ":" private@fujixerox.co.jp ":" capabilityMD1)
+nonce ":" cnonce ":" qop ":" MD(GET ":" "/object391"))
```

【0160】

上記の計算結果をresponse1とする。クライアント200は、HTTP  
オブジェクト391にアクセスするためには、以下のような要求メッセージを  
HTTPサーバ300に送信する。

【0161】

【数12】

```
GET /object391 HTTP/1.1
Authorization: Capability
    username=username
    realm="private@fujixerox.co.jp",
    nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
    uri="/object391",
    qop=auth,
    nc=00000001,
    cnonce="1bce22bf",
    capability=capability1,
    response=response1,
    opaque="a9f0171e7f40e415aa951069a403ecaf"
```

【0162】

HTTPサーバ300のアクセス制御オブジェクト301は、上記の要求メッ

セージ中のA u t h o r i z a t i o nフィールドから、以下の情報を取り出す。

- ユーザ情報
- 権限情報
- 権限情報のメッセージ・ダイジェスト

【0163】

さらに、アクセス制御オブジェクト301は、取り出したこれらの情報を基に、以下に示すアルゴリズム2（図4を参照のこと）を実行することによって、クライアント200がHTTPオブジェクト391に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【0164】

アルゴリズム2：

（ステップ11）A u t h o r i z a t i o nフィールドから抽出したユーザ名をu s e r i dとする。

【0165】

（ステップ12）A u t h o r i z a t i o nフィールドから抽出した権限情報をc a p a b i l i t yとする。

【0166】

（ステップ13）A u t h o r i z a t i o nフィールドから抽出された権限情報のメッセージ・ダイジェストをr e s p o n s eとする。

【0167】

（ステップ14）クライアント200に送信したメッセージ中のn o n c eフィールドの値をn o n c e 1とする。

【0168】

（ステップ15）u s e r i dに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをp a s s w o r dとする。他方、対応するパスワードが登録されていない場合は、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【0169】

(ステップ16) 一方向性関数の値MD (nonce1 + MD(capability + password)) を計算し、計算結果をdigestとする。

【0170】

(ステップ17) digestをresponseと比較する。両者が等しくない場合には、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【0171】

(ステップ18) capabilityを基に、クライアント200が要求するメソッド(すなわちオブジェクトに対する処理内容)、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求がcapabilityで許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【0172】

アルゴリズム2を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト301は、HTTPオブジェクト391に対してGETメソッドを起動するメッセージを送信する。

【0173】

図5には、本実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【0174】

トランザクションの前提として、クライアント100は、HTTPサーバ300に対するアクセスのアカウント(利用者資格)として、ユーザ情報"user id1"とパスワード"password1"を保持しているものとする。また、HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…についてのユーザ情報とパスワードの組み合わせ(user id1, password1), (user id2, password2) …を「パスワー

ド管理表」として保管している。

【0175】

クライアント100は、HTTPオブジェクト391へのアクセスに関し、クライアント200に与えてもよい権限情報 `capability1` を作成する (Tr11)。

【0176】

次いで、クライアント100は、権限情報 `capability1` の後ろに自己の秘密情報 `password1` をビット連結したビット列に一方方向性関数MDを適用して、保護化権限情報 `capabilityMD1` を作成する (Tr12)。

【0177】

次いで、クライアント100は、ユーザ情報 `userid1` と権限情報 `capability1`、及び保護化権限情報 `capabilityMD1` をクライアント200に送信することで、アクセス権限を安全に委譲する (Tr13)。

【0178】

アクセス権限を委譲されたクライアント200は、HTTPサーバ300に対してHTTPオブジェクト391へのアクセス要求を送信する (Tr14)。

【0179】

該アクセス要求には認証データが含まれていないので、HTTPサーバ300のアクセス制御オブジェクト301は、チャレンジ文字列 `nonce` を含んだ応答メッセージを送信する (Tr15)。

【0180】

クライアント200は、チャレンジ文字列 `nonce` と保護化権限情報 `capabilityMD1` をビット連結したビット列のメッセージ・ダイジェスト `response` を生成する (Tr16)。

【0181】

そして、クライアント200は、改めて、HTTPサーバ300に対してHTTPオブジェクト391へのアクセス要求を送信する (Tr17)。このアクセス要求には、クライアント100のユーザ名 `userid1`、権限情報 `capa`

b i l i t y 1、及びメッセージ・ダイジェスト r e s p o n s e を添付する。

【0182】

HTTPサーバ300のアクセス制御オブジェクト301は、クライアント200のアクセス要求の正当性を検証する(Tr18)。該検証は、上述のアルゴリズム1(図4を参照のこと)が規定する処理手順に従う。

【0183】

検証結果が成功裡に終了した場合には、アクセス制御オブジェクト301はアクセス要求を受理する(Tr19)。より具体的には、要求されているメソッド(例えばGETメソッド)を起動する。

【0184】

### 3. 第3の実施形態

次いで、本発明の第3の実施形態について説明する。第2の実施形態は、上記と同様、図1に示すような分散コンピューティング環境に適用可能である。但し、第3の実施形態は、上記の第1及び第2の実施形態とは相違し、権限情報に対して一方向性関数ではなく暗号化関数を適用することによってアクセス権限の安全な委譲を実現するものである。以下に示す例では、対称暗号鍵方式を用いた場合について説明するが、公開暗号鍵方式を用いても同様に本発明の効果を奏する点を充分理解されたい。

【0185】

本実施形態において、HTTPサーバ300は、" h t t p : / / w w w 3 0 0 " なるURLで表されるものとする。また、HTTPサーバ300は、アクセス制御オブジェクト301と、参照番号391~39Nで示されるN個のHTTPオブジェクトを所有している。各HTTPオブジェクト391~39Nはそれぞれ" h t t p : / / w w w 3 0 0 / o b j e c t 3 9 1 " , ... , " h t t p : / / w w w 3 0 0 / o b j e c t 3 9 N " なるURLで示されるものとする。

【0186】

アクセス制御オブジェクト301は、各HTTPオブジェクト391~39Nに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制

御オブジェクト301は、HTTPオブジェクト391～39Nと同じサーバ300上に存在する必要は必ずしもなく、例えば別のホスト（図示しない）上に存在してHTTPサーバ300から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【0187】

各クライアント100…は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント100は、ユーザ情報”userid1”とパスワード”password1”を保持しているものとする。

【0188】

HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…に関するユーザ情報とパスワードの組み合わせ（userid1, password1）,（userid2, password2）…を「パスワード管理表」（仮称）として保管している。

【0189】

また、アクセス制御オブジェクト301は、オブジェクトに対するアクセス権限を与えたクライアント・ユーザに対する「ケイパビリティ（capability）」を保持している。ケイパビリティとは、アクセス制御マトリックス（前述）を行毎すなわちユーザ毎に切り出した情報であり、ユーザが各HTTPオブジェクトに対して許容された操作権限を記述した情報である。

【0190】

各クライアント100…は、HTTPサーバ300に対して自己のユーザ情報とパスワードを提示することによって、自身に与えられたケイパビリティすなわち権限情報の範囲内でHTTPオブジェクト391…に対するアクセス操作が許容される。

【0191】

本実施形態では、HTTP1.1のBasic Authentication Scheme（基本認証法）に従ってHTTPメッセージが交換されるものとする。すなわち、クライアント100は、以下に示すようなHTTPリクエストを発行することにより、HTTPオブジェクト391のGETメソッドを起動

することができる。

【0192】

【数13】

GET /object391 HTTP/1.1

Authorization: Basic authdata

【0193】

但し、"authdata"は、ユーザ情報とパスワードを連結したビット列  
userid1:password1を、改行のないbase64エンコード法  
(前述)でエンコードした文字列である。

【0194】

次いで、クライアント100がクライアント200に権限情報を委譲する場合  
を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【0195】

権限情報は、以下の各事柄を含む文字列で構成される(但し、例示に過ぎず、  
使用許可回数など他の事柄を権限情報に含んでいてもよい)。

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時
- 有効期間終了日時

【0196】

ここで、クライアント100が、クライアント200にアクセス権限を委譲す  
るために、以下の権限情報文字列capability1を生成したとする。

【0197】

【数14】

((/object391), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)

【0198】

上記の権限情報文字列は、"/object391"で表されるHTTPオブ

ジェクト391に対して、GETメソッドのみを、世界時で1999年の4月24日10時0分48秒から1999年の4月25日10時0分48秒の間だけ実行することのできる権利を表している。

【0199】

本実施例では、クライアント200は、HTTPサーバ300のみを対象とするため、オブジェクト名としてサーバ内での名前のみを取り扱う。但し、一般には、オブジェクト名としてURL等のグローバルな識別子を用いることにより、複数のHTTPサーバのオブジェクトを識別するようにしてもよい。

#d

次いで、クライアント100は、自己の秘密情報password1を用いてこの権限情報capability1を暗号化して、以下に示す保護化権限情報capabilityCR1を生成する。

【0200】

【数15】

CRYPT(((/object), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT), password1)

【0201】

上記のうち暗号化関数CRYPTは、例えば、DES (Data Encryption Standard) やRC2、RC4、RC5などの任意の対称鍵暗号系を用いることができる。DESとは、米国政府が標準暗号として認定した共通鍵暗号方式のことである。鍵は56ビットの固定長（8ビットのパリティ・ビットを含めて64ビット）であり、まず56ビット長の鍵から48ビット長の鍵を16個作り、その鍵を使って64ビット長に分けたデータ・ブロックを16回だけ攪拌処理する。

【0202】

また、RC2、RC4及びRC5は、米RSAデータ・セキュリティ社が開発した共通鍵暗号方式である。このうち、RC2はブロック毎にデータを暗号化するブロック暗号であり、暗号鍵の長さを可変にできる。RC4は「ストリーム型

」と呼ぶ暗号方式を採用する。RC5は鍵長、データ・ブロック長ともに可変の暗号方式である。

【0203】

元の権限情報に対してこのような暗号化関数CRYPTを適用することによって、秘密情報password1を知らない第3者が保護化権限情報capabilityCR1を勝手に改竄することはできない。

【0204】

次いで、クライアント100は、少なくとも、

- userid1 (ユーザ情報)
- capabilityCR1 (保護化権限情報)

からなる情報を、他のクライアント200に対して送信する。

【0205】

既に述べたように、秘密情報password1を知らない第3者(クライアント200を含む)は、保護化権限情報capabilityCR1を改竄することができない。したがって、クライアント100は、他のクライアント200に対して、HTTPオブジェクト391に対するアクセス権限を安全に委譲することができる訳である。

【0206】

他方、上記の情報を受信したクライアント200は、クライアント100の代理として、HTTPオブジェクト391にアクセスするサービスを提供する。

【0207】

クライアント200は、HTTPオブジェクト391にアクセスするためには、少なくとも以下の情報を含む要求メッセージをHTTPサーバ300に送信する。

【0208】

- オブジェクト名
- メソッド名
- ユーザ名
- 暗号化された権限情報

【0209】

本実施例では、以下に示すような要求メッセージを送信する。

【0210】

【数16】

```
GET /object391 HTTP/1.1
Authorization: Capability cap1
```

【0211】

ここで、文字列 `cap1` は、ユーザ名と暗号化された権限情報とを” : ”を用いて連結して改行のない `base64` エンコード法（前述）でエンコードした文字列であるとする。

【0212】

HTTPサーバ300のアクセス制御オブジェクト301は、上記の要求メッセージ中の `Authorization` フィールドから、以下の情報を取り出す。

- ユーザ情報
- 暗号化された権限情報

【0213】

さらに、アクセス制御オブジェクト301は、取り出したこれらの情報を基に、以下に示すアルゴリズム3（図6を参照のこと）を実行することによって、クライアント200がHTTPオブジェクト391に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【0214】

アルゴリズム3：

（ステップ21） `Authorization` フィールドから抽出したユーザ名を `userid` とする。

【0215】

（ステップ22） `Authorization` フィールドから抽出した暗号化された権限情報を `capabilityCR` とする。

【0216】

(ステップ23) `userid`に相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードを`password`とする。他方、対応するパスワードが登録されていない場合は、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【0217】

(ステップ24) 暗号化された権限情報`capabilityCR`を、秘密情報`password`を用いて復号化する。復号化結果を`capability`とする。

【0218】

(ステップ25) 復号化された権限情報`capability`が正しい文法規則に則っているか否かを判断する。正しくない場合には、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。正しい場合には次ステップへ進む。

【0219】

(ステップ26) 復号化された`capability`を基に、クライアント200が要求するメソッド(すなわちオブジェクトに対する処理内容)、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求が`capability`で許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【0220】

アルゴリズム3を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト301は、HTTPオブジェクト391に対してGETメソッドを起動するメッセージを送信する。

【0221】

図7には、本実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【0222】

トランザクションの前提として、クライアント100は、HTTPサーバ300に対するアクセスのアカウント（利用者資格）として、ユーザ情報” user id1” とパスワード” password1” を保持しているものとする。また、HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…についてのユーザ情報とパスワードの組み合わせ（user id1, password1）,（user id2, password2）…を「パスワード管理表」として保管している。

【0223】

クライアント100は、HTTPオブジェクト391へのアクセスに関し、クライアント200に与えてもよい権限情報 capability1を作成する（Tr21）。

【0224】

次いで、クライアント100は、自己の秘密情報 password1を用いて権限情報 capability1を暗号化して、保護化権限情報 capabilityCR1を作成する（Tr22）。

【0225】

次いで、クライアント100は、保護化権限情報 capabilityCR1をクライアント200に送信することで、アクセス権限を安全に委譲する（Tr23）。

【0226】

アクセス権限を委譲されたクライアント200は、HTTPサーバ300に対してHTTPオブジェクト391へのアクセス要求を送信する（Tr24）。このアクセス要求には、クライアント100のユーザ名 user id1と保護化権限情報 capabilityCR1を添付する。

【0227】

HTTPサーバ300のアクセス制御オブジェクト301は、クライアント200のアクセス要求の正当性を検証する（Tr25）。該検証は、上述のアルゴリズム3（図6を参照のこと）が規定する処理手順に従う。

【0228】

検証結果が成功裡に終了した場合には、アクセス制御オブジェクト301はアクセス要求を受理する(Tr26)。より具体的には、要求されているメソッド(例えばGETメソッド)を起動する。

【0229】

#### 4. 第4の実施形態

次いで、本発明の第4の実施形態について説明する。第4の実施形態は、上記と同様、図1に示すような分散コンピューティング環境に適用可能である。第4の実施形態は、権限情報を暗号化するとともに、クライアント-HTTPサーバ間のアクセス要求の際に「チャレンジ・レスポンス認証」(前述)を用いることによってアクセス権限の安全な委譲を実現するものである。

【0230】

本実施形態において、HTTPサーバ300は、"http://www300"なるURLで表されるものとする。また、HTTPサーバ300は、アクセス制御オブジェクト301と、参照番号391~39Nで示されるN個のHTTPオブジェクトを所有している。各HTTPオブジェクト391~39Nはそれぞれ"http://www300/object391", ..., "http://www300/object39N"なるURLで示されるものとする。

【0231】

アクセス制御オブジェクト301は、各HTTPオブジェクト391~39Nに対するアクセス要求を制御するためのオブジェクトである。但し、アクセス制御オブジェクト301は、HTTPオブジェクト391~39Nと同じサーバ300上に存在する必要は必ずしもなく、例えば別のサーバ(図示しない)上に存在してHTTPサーバ300から遠隔的に呼び出されて所定のアクセス制御処理を起動するようにしてもよい。

【0232】

各クライアント100...は、ユーザを識別可能なユーザ情報と、秘密情報としてのパスワードを保持している。クライアント100は、ユーザ情報"user id1"とパスワード"password1"を保持しているものとする。

【0233】

HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…に関するユーザ情報とパスワードの組み合わせ (userid1, password1), (userid2, password2) …を「パスワード管理表」(仮称)として保管している。

【0234】

また、アクセス制御オブジェクト301は、オブジェクトに対するアクセス権限を与えたクライアント・ユーザに対する「ケイパビリティ (capability)」を保持している。ケイパビリティとは、アクセス制御マトリックス(前述)を行毎すなわちユーザ毎に切り出した情報であり、ユーザが各HTTPオブジェクトに対して許容された操作権限を記述した情報である。

【0235】

各クライアント100…は、HTTPサーバ300に対して自己のユーザ情報とパスワードを提示することによって、自身に与えられたケイパビリティすなわち権限情報の範囲内でHTTPオブジェクト391…に対するアクセス操作が許容される。

【0236】

本実施形態では、HTTP1.1のDigest Authentication Scheme (基本認証法)に従ってHTTPメッセージが交換されるものとする。すなわち、クライアント100は、以下に示すようなHTTPリクエストを発行することにより、HTTPオブジェクト391のGETメソッドを起動することができる。

【0237】

【数17】

```
GET /object391 HTTP/1.1
Authorization: Digest digest-response
```

【0238】

但し、"digest-response"は、HTTP1.1の規定に基

づき生成された認証データである。

【0239】

次いで、クライアント100がクライアント200に権限情報を委譲する場合を例にとって、本実施形態に係るアクセス権限委譲方法について説明する。

【0240】

権限情報は、以下の各事柄を含む文字列で構成される（但し、例示に過ぎず、使用許可回数など他の事柄を権限情報に含んでいてもよい）。

- オブジェクト名の列
- メソッド名の列
- 有効期間開始日時
- 有効期間終了日時

【0241】

ここで、クライアント100が、クライアント200にアクセス権限を委譲するために、以下の権限情報文字列 `capability1` を生成したとする。

【0242】

【数18】

`((/object391), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT)`

【0243】

上記の権限情報文字列は、“`/object391`”で表されるHTTPオブジェクト391に対して、GETメソッドのみを、世界時で1999年の4月24日10時0分48秒から1999年の4月25日10時0分48秒の間だけ実行することのできる権利を表している。

【0244】

次いで、クライアント100は、自己の秘密情報 `password1` を用いてこの権限情報 `capability1` を暗号化して、以下に示す保護化権限情報 `capabilityCR1` を生成する。

【0245】

【数 19】

CRYPT(((/object), (GET), Apr:24:10:00:48:1999:GMT,  
Apr:25:10:00:48:1999:GMT), password1)

【0246】

上記のうち暗号化関数CRYPTは、例えば、DES (Data Encryption Standard) やRC2、RC4、RC5などの任意の対称鍵暗号系（前述）を用いることができる。元の権限情報に対してこのような暗号化を適用することによって、秘密情報password1を知らない第3者が保護化権限情報capabilityCR1を勝手に改竄することはできない。

【0247】

次いで、クライアント100は、少なくとも、

- userid1 (ユーザ情報)
- capability1 (権限情報)
- capabilityCR1 (保護化権限情報)

からなる情報を、他のクライアント200に対して送信する。

【0248】

既に述べたように、秘密情報password1を知らない第3者（クライアント200を含む）は、保護化権限情報capabilityCR1を改竄することができない。したがって、クライアント100は、他のクライアント200に対して、HTTPオブジェクト391に対するアクセス権限を安全に委譲することができる訳である。

【0249】

他方、上記の情報を受信したクライアント200は、クライアント100の代理として、HTTPオブジェクト391にアクセスするサービスを提供する。

【0250】

クライアント200は、HTTPオブジェクト391にアクセスするために、まず、以下に示すようなメッセージをHTTPサーバ300に送信する。

【0251】

【数 20】

GET /object391 HTTP/1.1

【0252】

この場合、要求メッセージに認証データが含まれていないため、HTTPサーバ300は、以下のような応答メッセージをクライアント200に送信する。

【0253】

【数 21】

HTTP/1.1 401 Unauthorized

WWW-Authenticate: Capability

realm="private@fujixerox.co.jp",

qop="auth, auth-int",

nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",

opaque="a9f0171e7f40e415aa951069a403ecaf"

【0254】

上記のうち、nonceに代入された文字列は、「チャレンジ・レスポンス認証」を行うための「チャレンジ文字列」（前述）である。

【0255】

クライアント200は、保護化権限情報capabilityCR1を用いてチャレンジ文字列nonceを暗号化した値を計算する。この場合の例で言えば、以下の式を計算することになる。すなわち、

【0256】

【数 22】

CRYPT("0a093dad98c11d0f600cfc7102dd2f0e8c", capabilityCR1)

【0257】

また、その他の文字列もさらに連結するようにしてもよい。例えば、HTTP/1.1のように以下の式を計算してもよい。

【 0 2 5 8 】

【数 2 3】

```
CRYPT(CRYPT(userid ":" private@fujixerox.co.jp ":"
capabilityCRI)+nonce ":" cnonce ":" qop ":"
CRYPT(GET ":" "/object391"))
```

【 0 2 5 9 】

上記の計算結果を response1 とする。クライアント 200 は、HTTP オブジェクト 391 にアクセスするためには、以下のような要求メッセージを HTTP サーバ 300 に送信する。

【 0 2 6 0 】

【数 2 4】

```
GET /object391 HTTP/1.1
Authorization: Capability
    username=username
    realm="private@fujixerox.co.jp",
    nonce="0a093dad98c11d0f600cfc7102dd2f0e8c",
    uri="/object391",
    qop=auth,
    nc=00000001,
    cnonce="1bce22bf",
    capability=capability1,
    response=response1,
    opaque="a9f0171e7f40e415aa951069a403ecaf"
```

【 0 2 6 1 】

HTTP サーバ 300 のアクセス制御オブジェクト 301 は、上記の要求メッセージ中の Authorization フィールドから、以下の情報を取り出す。

● ユーザ情報

- 権限情報
- 権限情報を暗号化した値

【0262】

さらに、アクセス制御オブジェクト301は、取り出したこれらの情報を基に、以下に示すアルゴリズム4（図8を参照のこと）を実行することによって、クライアント200がHTTPオブジェクト391に対するアクセス権限を正当に委譲された者か否かを判断することができる。

【0263】

アルゴリズム4：

（ステップ31）Authorizationフィールドから抽出したユーザ名をuseridとする。

【0264】

（ステップ32）Authorizationフィールドから抽出した権限情報をcapabilityとする。

【0265】

（ステップ33）Authorizationフィールドから抽出された権限情報の暗号化値をresponseとする。

【0266】

（ステップ34）クライアント200に送信したメッセージ中のnonceフィールドの値をnonce1とする。

【0267】

（ステップ35）useridに相当するパスワードをパスワード管理表から取得する。該管理表から取得されたパスワードをpasswordとする。他方、対応するパスワードが登録されていない場合は、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。

【0268】

（ステップ36）暗号化関数の値CRYPT(nonce1, CRYPT(capability, password))を計算し、計算結果をdigestとする。

【0269】

(ステップ37) `digest`を`response`と比較する。両者が等しくない場合には、クライアント200からの要求メッセージは不正アクセスであるとして、本アルゴリズムを終了する。両者が一致する場合には次ステップへ進む。

【0270】

(ステップ38) `capability`を基に、クライアント200が要求するメソッド(すなわちオブジェクトに対する処理内容)、対象オブジェクト、有効期限などの、権限の有効性をチェックする。要求が`capability`で許可されないものであれば、不正アクセスであるとして本アルゴリズムを終了する。許可されている範囲内であれば、正当なアクセスであるとして本アルゴリズムを終了する。

【0271】

アルゴリズム4を実行した結果、正当アクセスであることが判明した場合には、アクセス制御オブジェクト301は、HTTPオブジェクト391に対してGETメソッドを起動するメッセージを送信する。

【0272】

図9には、本実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解している。以下、同図を参照しながら説明する。

【0273】

トランザクションの前提として、クライアント100は、HTTPサーバ300に対するアクセスのアカウント(利用者資格)として、ユーザ情報”`userid1`”とパスワード”`password1`”を保持しているものとする。また、HTTPサーバ300のアクセス制御オブジェクト301は、各クライアント100…についてのユーザ情報とパスワードの組み合わせ(`userid1`, `password1`), (`userid2`, `password2`)…を「パスワード管理表」として保管している。

【0274】

クライアント100は、HTTPオブジェクト391へのアクセスに関し、ク

クライアント 200 に与えてもよい権限情報 `capability1` を作成する (Tr 31)。

【0275】

次いで、クライアント 100 は、自己の秘密情報 `password1` を用いて権限情報 `capability1` を暗号化して、保護化権限情報 `capability1 CRYPT1` を作成する (Tr 32)。

【0276】

次いで、クライアント 100 は、ユーザ情報 `userid1` と権限情報 `capability1`、及び保護化権限情報 `capability1 CRYPT1` をクライアント 200 に送信することで、アクセス権限を委譲する (Tr 33)。

【0277】

アクセス権限を委譲されたクライアント 200 は、HTTP サーバ 300 に対して HTTP オブジェクト 391 へのアクセス要求を送信する (Tr 34)。

【0278】

該アクセス要求には認証データが含まれていないので、HTTP サーバ 300 のアクセス制御オブジェクト 301 は、チャレンジ文字列 `nonce` を含んだ応答メッセージを送信する (Tr 35)。

【0279】

クライアント 200 は、保護化権限情報 `capability1 CR1` を用いてチャレンジ文字列 `nonce` を暗号化して、`response` を生成する (Tr 36)。

【0280】

そして、クライアント 200 は、改めて、HTTP サーバ 300 に対して HTTP オブジェクト 391 へのアクセス要求を送信する (Tr 37)。このアクセス要求には、クライアント 100 のユーザ名 `userid1`、権限情報 `capability1`、及び `response` を添付する。

【0281】

HTTP サーバ 300 のアクセス制御オブジェクト 301 は、クライアント 200 のアクセス要求の正当性を検証する (Tr 38)。該検証は、上述のアルゴ

リズム 1 (図 8 を参照のこと) が規定する処理手順に従う。

【0282】

検証結果が成功裡に終了した場合には、アクセス制御オブジェクト 301 はアクセス要求を受理する (Tr 39)。より具体的には、要求されているメソッド (例えば GET メソッド) を起動する。

【0283】

5. 保護化権限情報の安全な保存方法

上述した第 1 ~ 第 4 の実施形態によれば、クライアント 100 は、他のクライアント 200 に対して HTTP オブジェクトへのアクセス権限を委譲するに際し、自らのパスワード password1 をそのまま渡したり、クライアント 200 に与える権限情報 capability1 そのものを送信することはせず、権限情報 capability1 に対して一方向性関数 MD や暗号化関数 CRYPT を適用して、権限情報を不正に改竄できないような「保護化権限情報」の形態にしてから委譲するようにした。

【0284】

この結果、権限情報の内容を勝手に書き換えて HTTP オブジェクトに不正アクセスされる機会 (例えば有効期限を無断で延長したり、有効使用回数を増やしたり、オブジェクトへの操作権限を強化したすすること) を、好適に排除することができる。

【0285】

しかしながら、保護された形式であれ、権限情報を複製することで、権限を正当に譲受していない者さえ HTTP オブジェクトにアクセスすることが可能になってしまう。

【0286】

例えば、図 10 に示すように、クライアント 100 から正当にアクセス権限を譲受したクライアント 200 が、自らは権限情報を正当にのみ使用していたとしても、保護化権限情報を格納した 2 次記憶装置 (例えばハード・ディスク装置) 201 が攻撃を受けることで、保護化権限情報は漏洩し、悪用を許してしまいかねない。メモリ空間すなわち揮発的なメモリ上にロードされた保護化権限情報を

プロテクトすることは比較的容易であるが、ファイル空間すなわちローカル・ディスク上の格納された情報をプロテクトすることは容易ではない。特に、分散コンピューティング環境では、リモートのホスト装置間で互いのローカル・ディスクはトランスパレントな状態なので、攻撃を受け易い。

【0287】

以下では、クライアント200が委譲された保護化権限情報をローカル・ディスクに安全且つ不揮発的に格納する方法について説明する。

【0288】

#### 5-1. 方法1

まず、クライアント100は、他のクライアント200に保護化権限情報を送信する。

【0289】

次いで、クライアント100は、クライアント200に対して暗号化鍵を送信する。

【0290】

クライアント200は、保護化権限情報を自身のローカル・ディスク201に格納するときには、受信した暗号化鍵を用いて暗号化してから行う。この結果、不正クライアントがローカル・ディスク201に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【0291】

クライアント200がHTTPオブジェクト391へのアクセス要求を行うときには、ローカル・ディスク201から保護化権限情報を取り出すとともに、暗号化鍵で復号化して用いればよい。保護化権限情報はクライアント200のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【0292】

HTTPサーバ300に対するアクセス要求及び認証の処理手順は上記に従う。

【0293】

## 5-2. 方法2

まず、クライアント100は、暗号化鍵を用いて保護化権限情報を暗号化して、2重に保護化された権限情報を生成する。

【0294】

次いで、クライアント100は、他のクライアント200に、2重に保護化された権限情報を送信する。

【0295】

そして、クライアント200は、受け取った保護化権限情報をローカル・ディスク201にそのまま格納する。この保護化権限情報は二重に保護され、そのままアクセス要求に使用することはできない。したがって、不正クライアントがローカル・ディスク201に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【0296】

クライアント200がオブジェクトのアクセス要求を発行したいときには、2重に保護化された権限情報を復号化するための復号化鍵を、クライアント100から受け取ればよい。

【0297】

そして、クライアント200は、受け取った復号化鍵を用いて2重に保護化された権限情報を復号化することで、利用可能な形式の保護化権限情報を得ることができる。保護化権限情報はクライアント200のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【0298】

HTTPサーバ300に対するアクセス要求及び認証の処理手順は上記に従う。

【0299】

なお、暗号化鍵と復号化鍵は、対称鍵暗号方式における同一鍵であっても、あるいは公開鍵暗号方式における秘密鍵と公開鍵の組み合わせであってもよい。

【0300】

## 5-3. 方法3

まず、クライアント 100 は、他のクライアント 200 に保護化権限情報を送信する。

【0301】

クライアント 200 は、情報を暗号化するための暗号化鍵と、暗号化鍵で暗号化された暗号化情報を復号化するための復号化鍵とを用意しておく。暗号化鍵と復号化鍵は、公開鍵暗号方式における秘密鍵と公開鍵の組み合わせでよい。

【0302】

クライアント 200 は、復号化鍵をクライアント 100 に送信する。復号化鍵をローカル・ディスク 201 に格納してしまうと、不正クライアントがローカル・ディスク 201 に攻撃を仕掛けて、外部に漏洩する危険がある。このため、クライアント 200 は、クライアント 100 に復号化鍵を送信した後、直ちに復号化鍵を廃棄しておくことが好ましい。

【0303】

そして、クライアント 200 は、保護化権限情報をさらに暗号化鍵で暗号化してから、ローカル・ディスクに格納する。この結果、不正クライアントがローカル・ディスク 201 に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

【0304】

クライアント 200 は、オブジェクトのアクセス要求を発行したいときには、2重に保護化された権限情報をローカル・ディスク 201 から取り出すとともに、復号化鍵を用いてこれを復号化して、元の保護化権限情報を再現すればよい。但し、クライアント 200 が復号化鍵を廃棄してしまっている場合には、クライアント 100 から復号化鍵を改めて取り寄せればよい。保護化権限情報はクライアント 200 のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

【0305】

HTTPサーバ 300 に対するアクセス要求及び認証の処理手順は上記に従う。

【0306】

#### 5-4. 方法4

まず、クライアント100は、他のクライアント200に保護化権限情報を送信する。

##### 【0307】

次いで、クライアント200は、クライアント100に「チャレンジ・レスポンス認証」（前述）を行うためのチャレンジ文字列を送信する。

##### 【0308】

次いで、クライアント100は、受信したチャレンジ文字列と所定の秘密情報に対して所定の演算処理を適用して暗号鍵を生成する。そして、クライアント100は、生成した暗号鍵をクライアント200に送信する。

##### 【0309】

これに対し、クライアント200は、受信した暗号鍵を用いて保護化権限情報を暗号化してから、ローカル・ディスク201に格納する。この結果、不正クライアントがローカル・ディスク201に攻撃を仕掛けても、保護化権限情報は暗号化されているので、万一漏洩しても悪用される心配がない。

##### 【0310】

クライアント200がHTTPオブジェクト391へのアクセス要求を行うときには、ローカル・ディスク201から保護化権限情報を取り出すとともに、暗号化鍵で復号化して用いればよい。保護化権限情報はクライアント200のメモリ上にしか存在しないのでセキュリティ管理が容易となる点を充分理解されたい。

##### 【0311】

HTTPサーバ300に対するアクセス要求及び認証の処理手順は上記に従う。

##### 【0312】

なお、上記の各方法では、保護化権限情報を安全且つ不揮発的にローカル・ディスク201に格納する場合を例にとって説明したが、格納する情報は保護化権限情報に限定されず、他の秘密性の高い情報（例えばユーザのパスワードなど）の格納にも応用することができる。

【0313】

[追補]

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【0314】

なお、URL (Uniform Resource Locator) については、例えばRFC (Request For Comments) 1738やRFC1808などに記述されている。また、HTML (Hyper Text Markup Language) については、例えばRFC1866に記述されている。また、HTTP (Hyper Text Transfer Protocol) については、例えばRFC1945やRFC2068に記述されている。

【0315】

【発明の効果】

以上詳記したように、本発明によれば、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」をサブジェクト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することができる。

【0316】

また、本発明によれば、複数のホストがネットワーク接続され且つネットワーク上にオブジェクトが散在する分散コンピューティング環境において、オブジェクトに関するアクセス権限を記述した「ケイパビリティ」を、ホスト（ユーザ）間で安全に転送することができる、優れたアクセス権限委譲方法を提供することができる。

【0317】

また、本発明によれば、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、他のサブジェクトに安全に委譲

することができる、優れたアクセス権限委譲方法を提供することができる。

【0318】

また、本発明によれば、ケイパビリティを保有するサブジェクトが権限内容を変更したケイパビリティを自由に生成し、且つ、生成されたケイパビリティをオブジェクトの管理者が安全に検査することができる、優れたアクセス権限委譲方法を提供することができる。

【図面の簡単な説明】

【図1】

本発明の実施形態に係る分散コンピューティング環境を模式的に示した図である。

【図2】

本発明の第1の実施形態において、アクセス権限オブジェクト301がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図3】

本発明の第1の実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解していたものである。

【図4】

本発明の第2の実施形態において、アクセス権限オブジェクト301がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図5】

本発明の第2の実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解していたものである。

【図6】

本発明の第3の実施形態において、アクセス権限オブジェクト301がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図7】

本発明の第3の実施形態において、クライアント100、クライアント200

及びHTTPサーバ300間で実行されるトランザクションを図解していたものである。

【図8】

本発明の第4の実施形態において、アクセス権限オブジェクト301がアクセスの正当性を判断するための処理手順を示したフローチャートである。

【図9】

本発明の第4の実施形態において、クライアント100、クライアント200及びHTTPサーバ300間で実行されるトランザクションを図解していたものである。

【図10】

本発明の実施形態に係る分散コンピューティング環境を模式的に示した図であり、より具体的にはクライアント200の2次記憶装置に格納した保護化権限情報が不正利用される様子を図解したものである。

【符号の説明】

100, 200…クライアント

300…HTTPサーバ

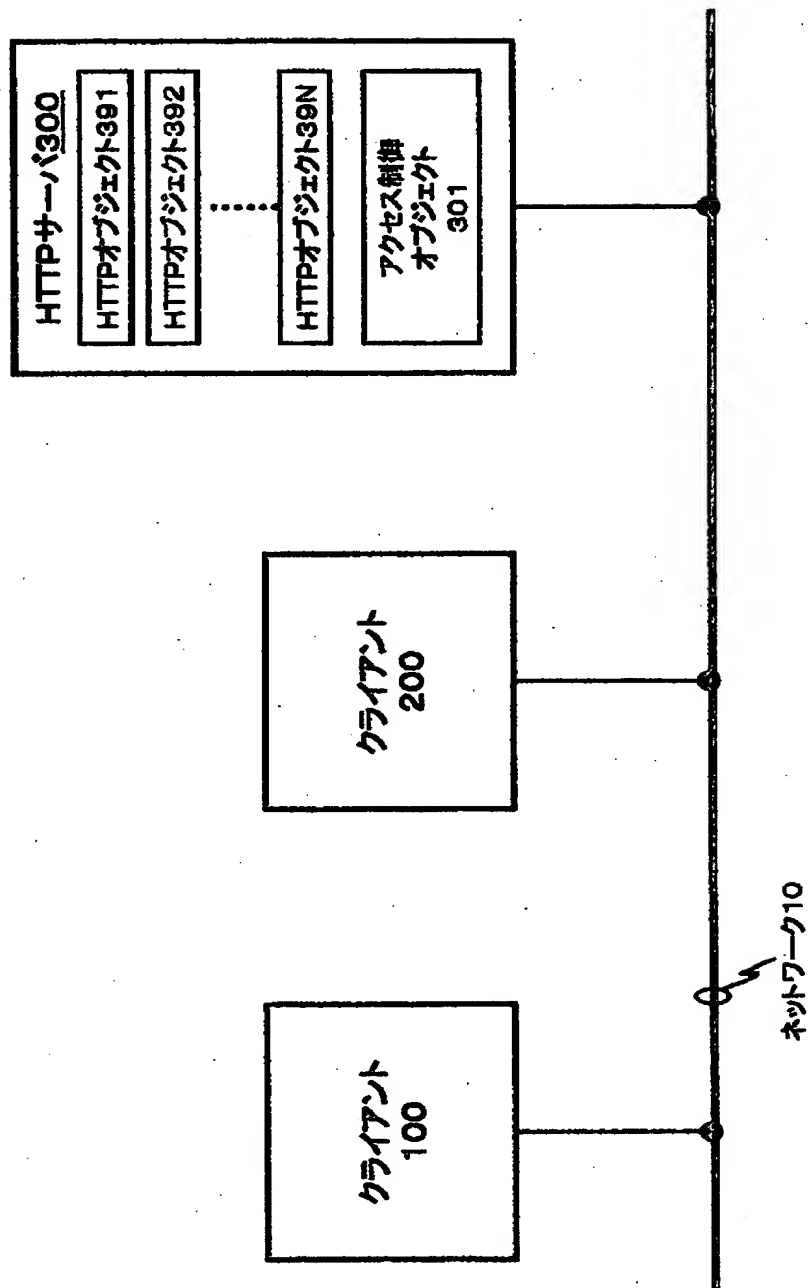
301…アクセス制御オブジェクト

391～39N…HTTPオブジェクト

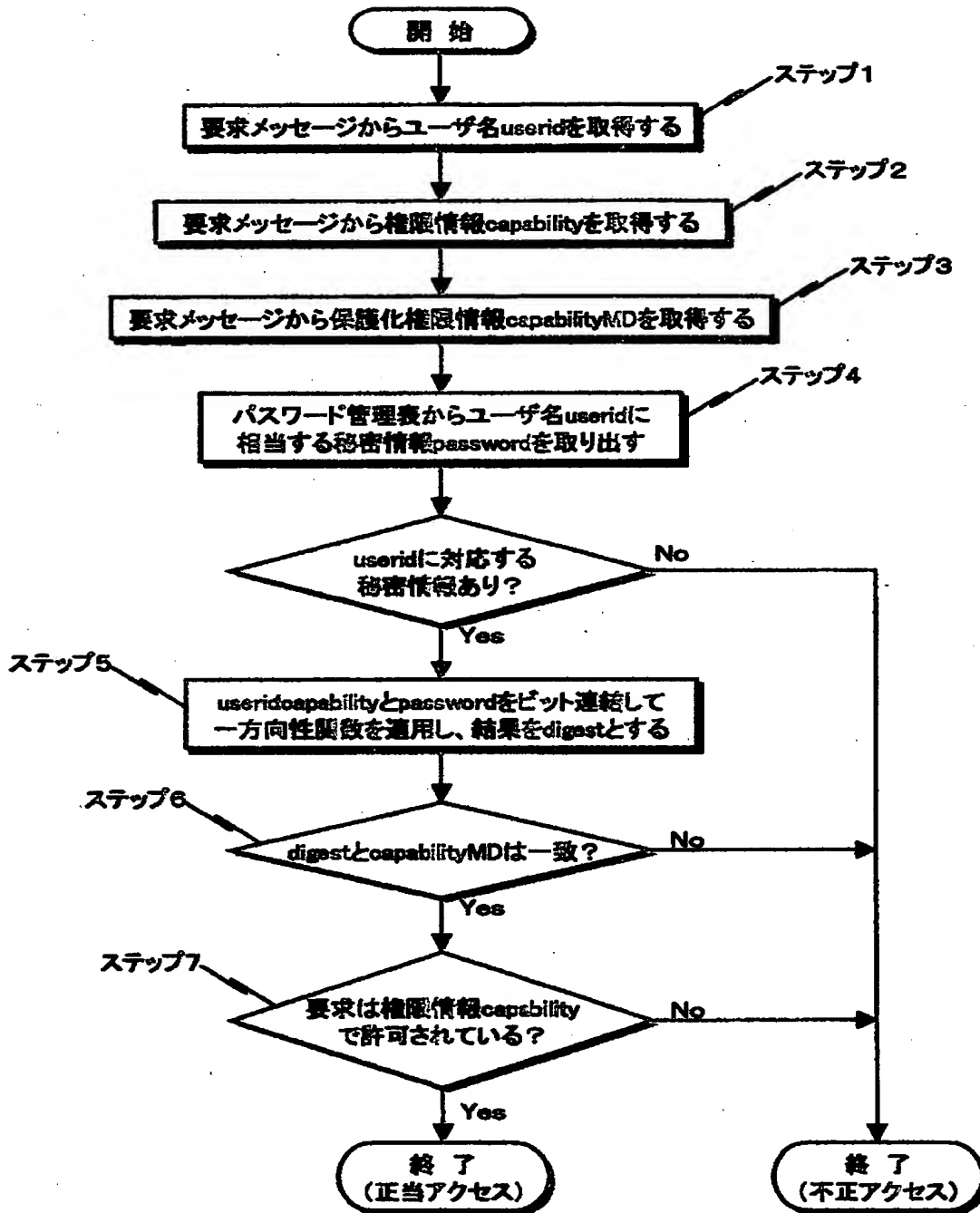
【書類名】

図面

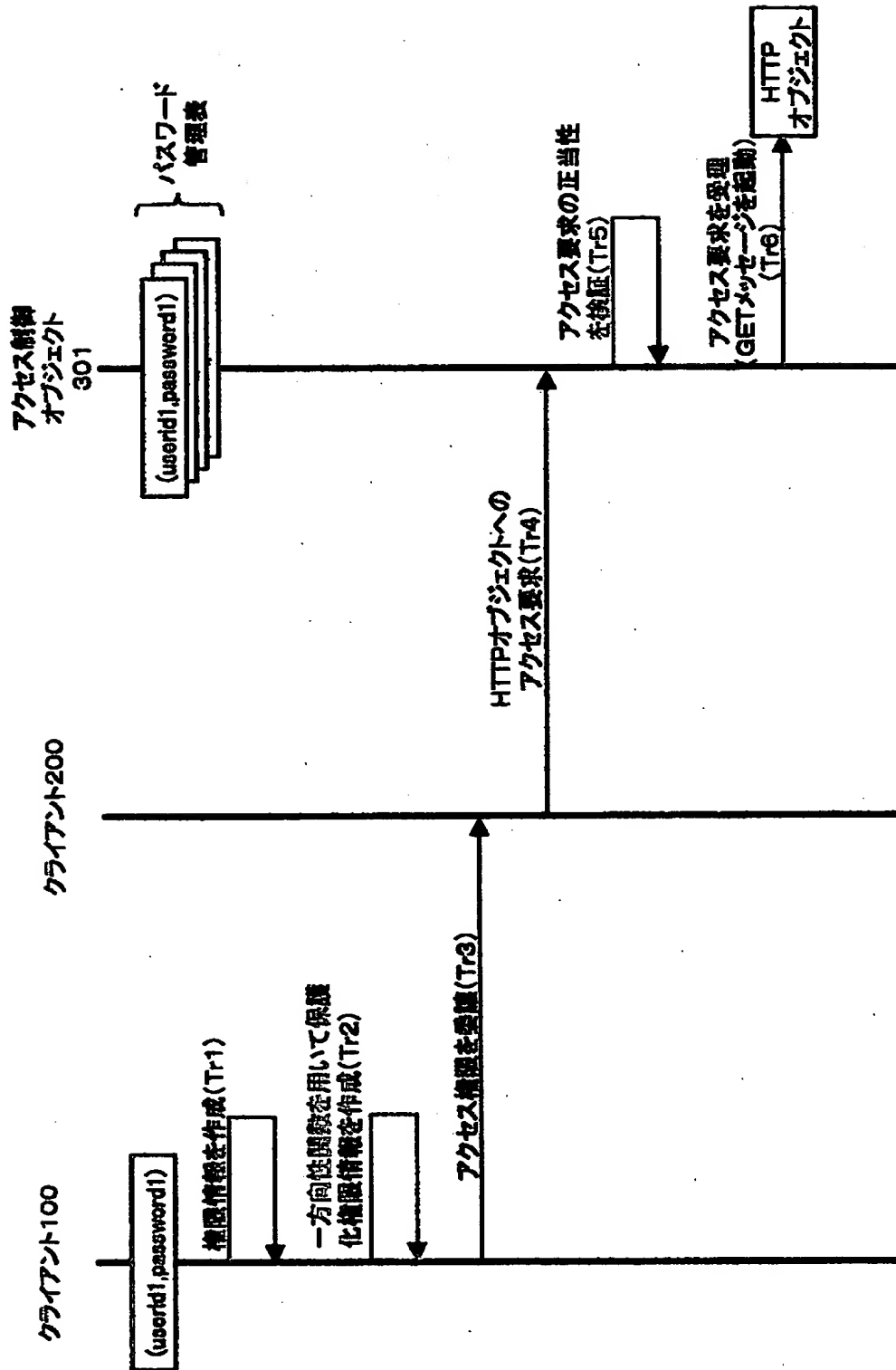
【図 1】



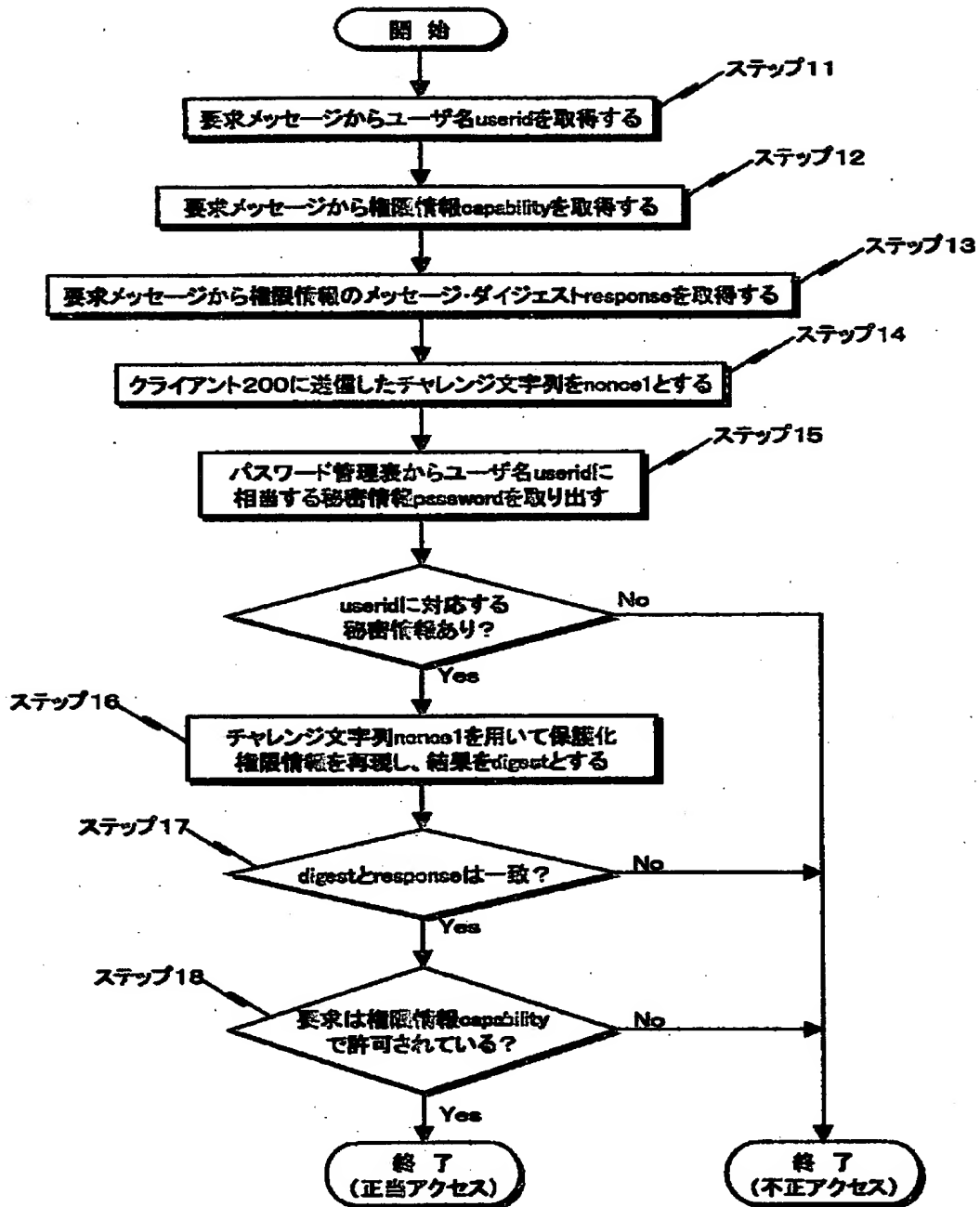
【図 2】



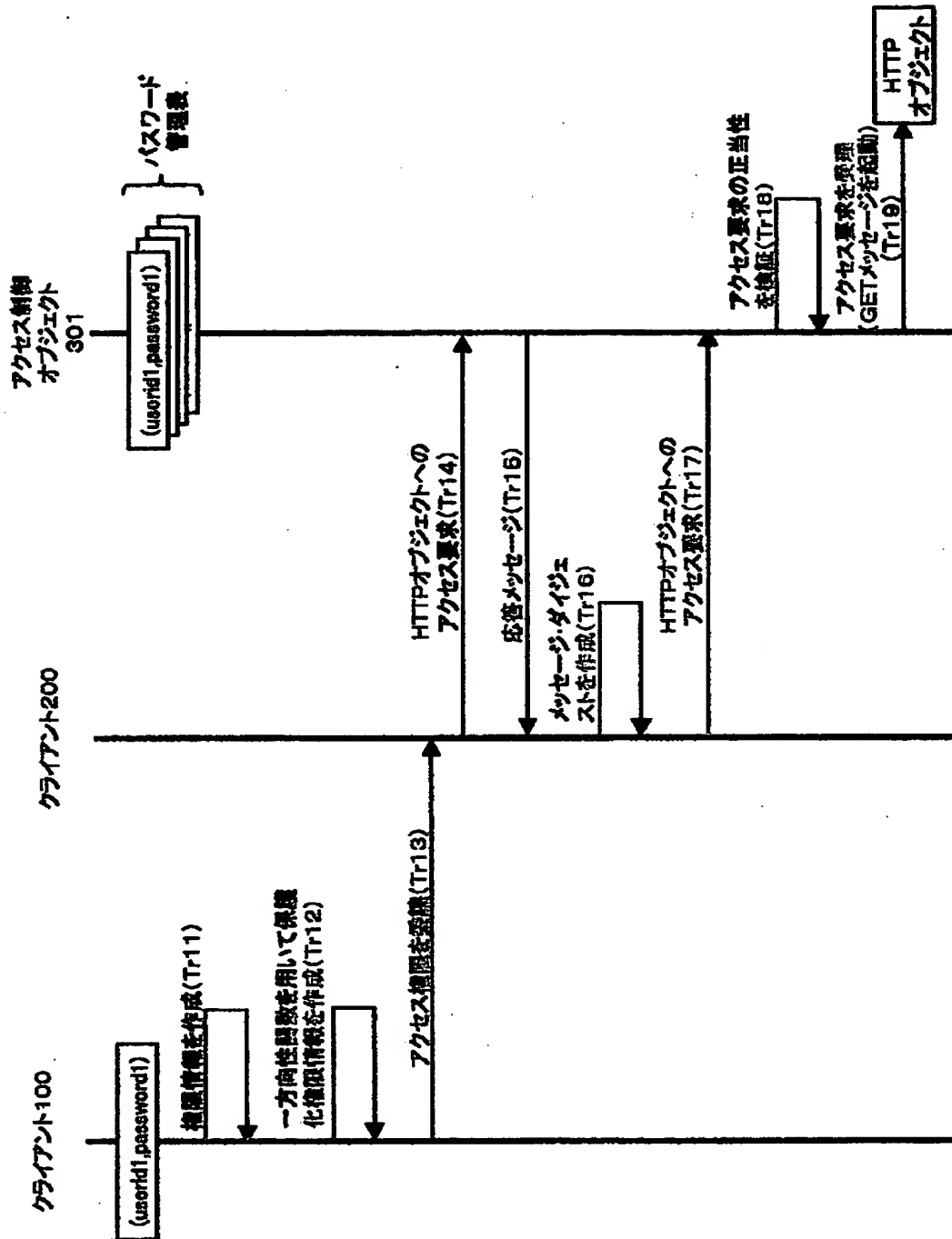
【図 3】



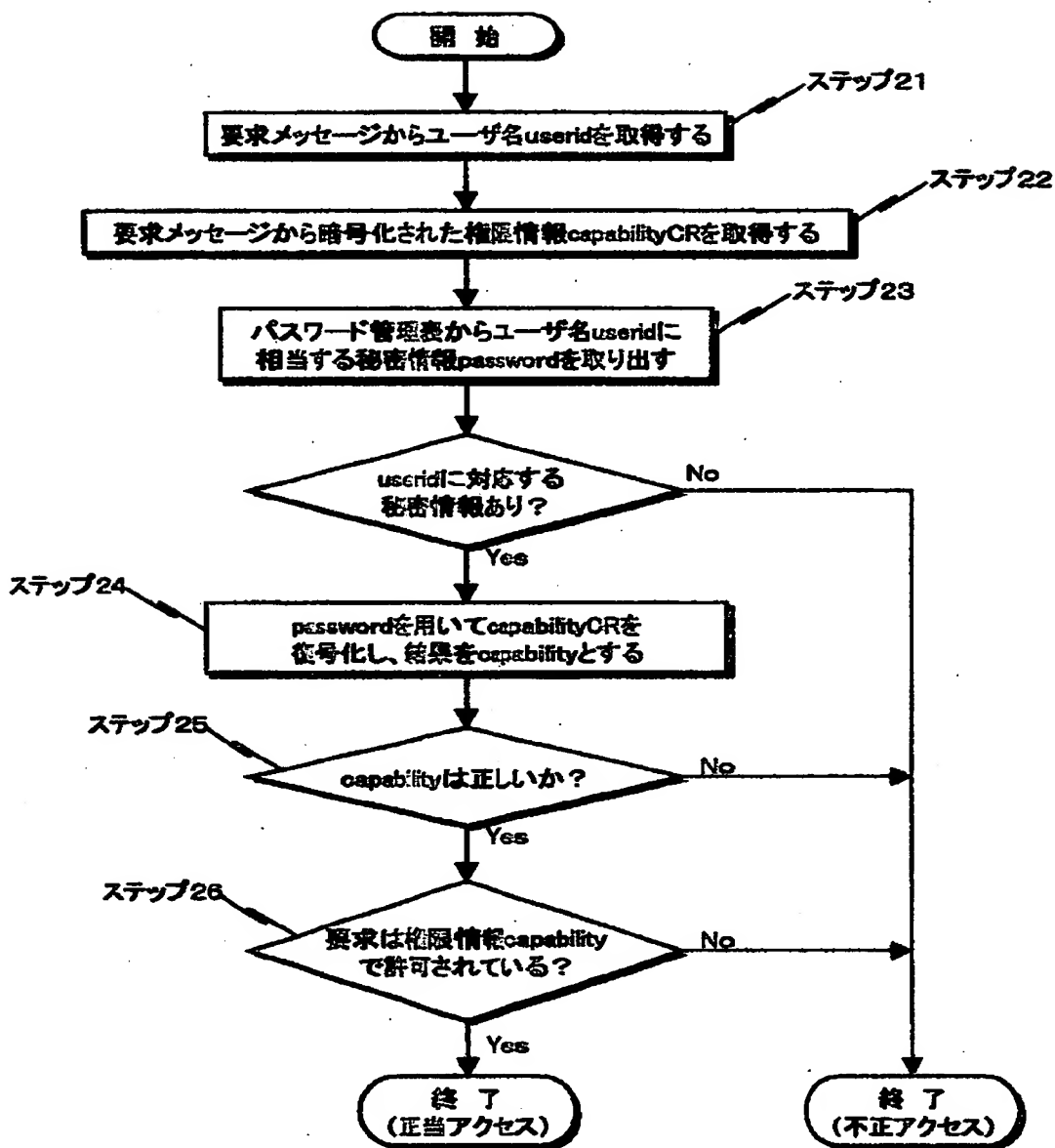
【図 4】



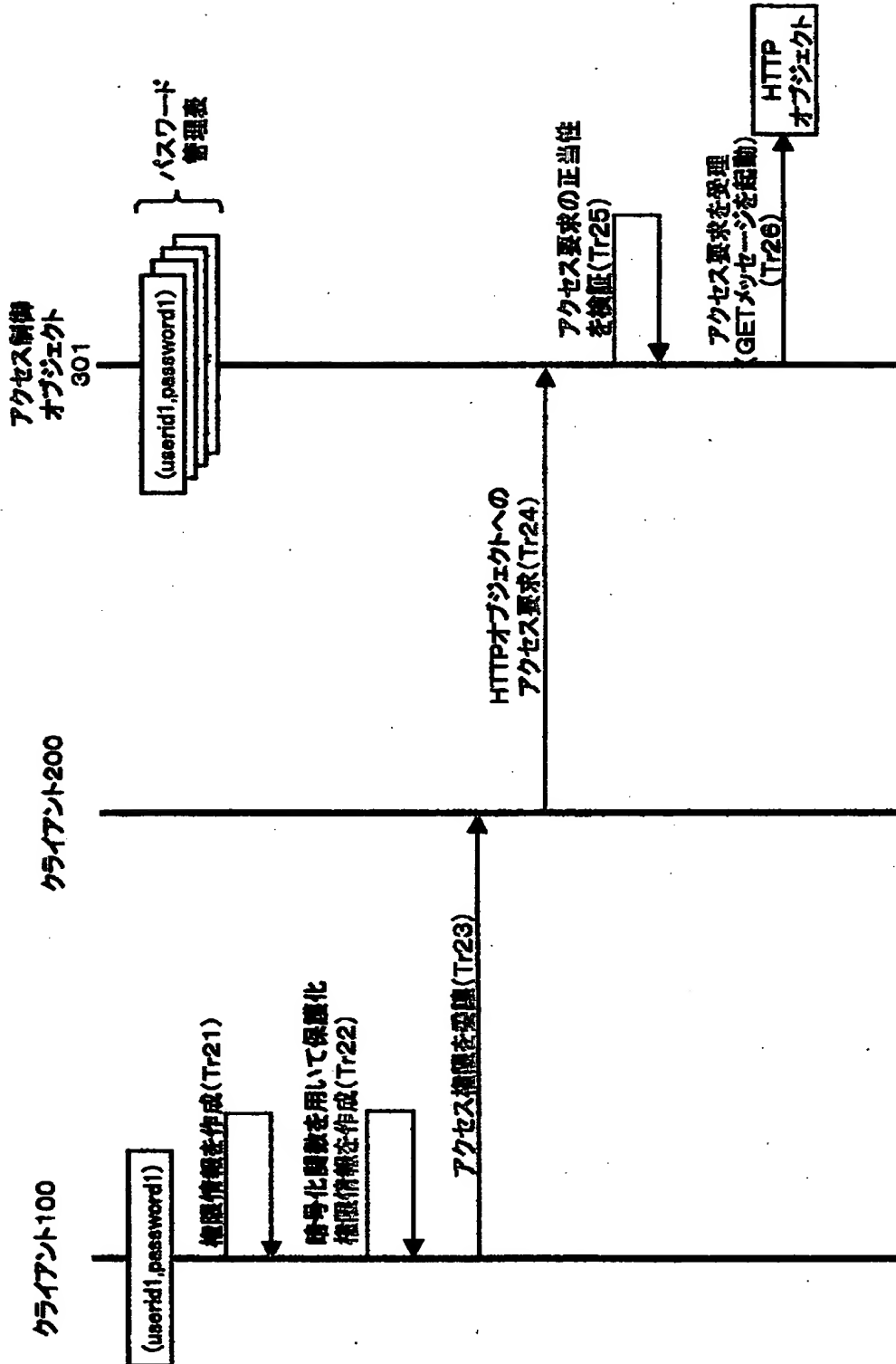
【図 5】



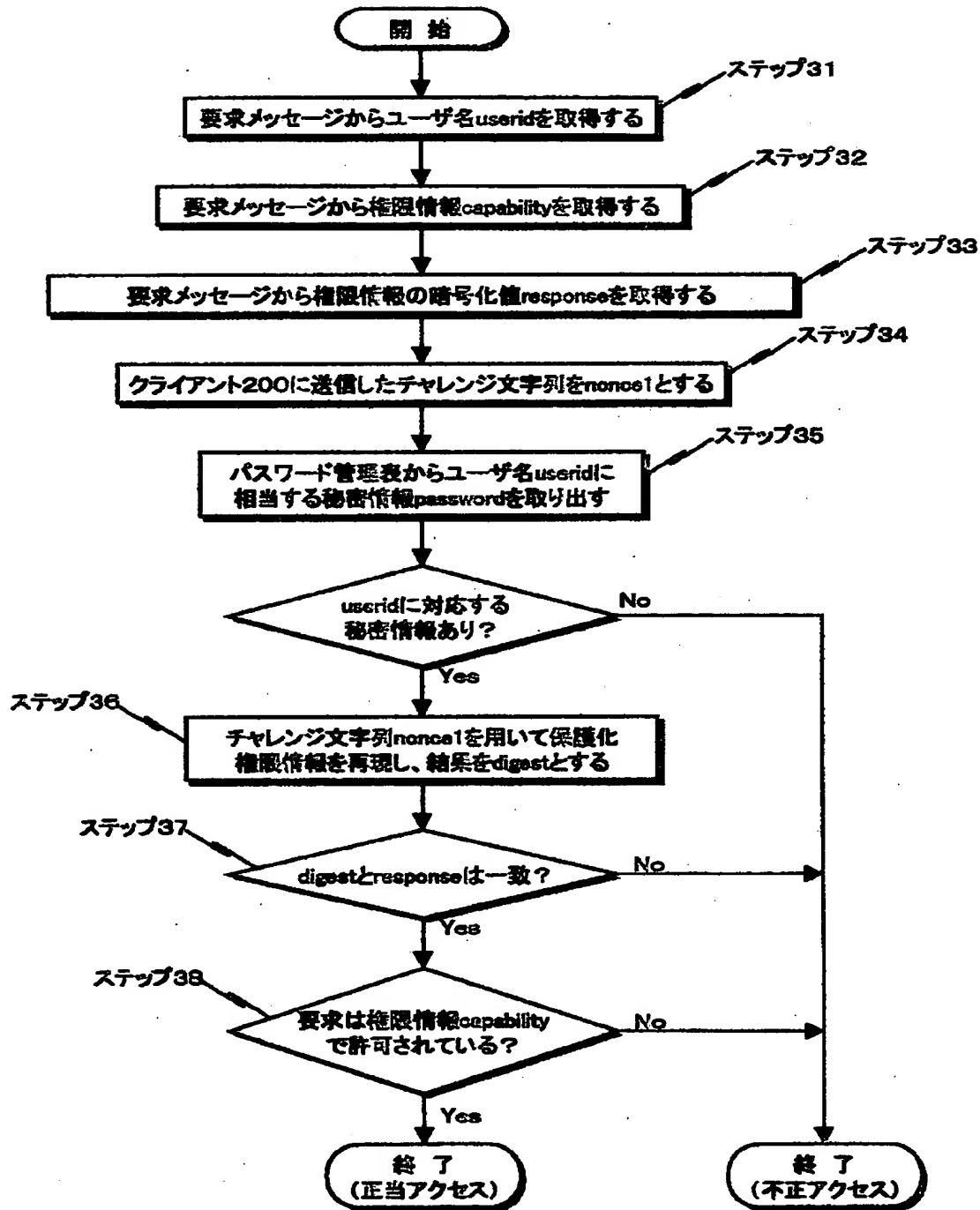
【図 6】



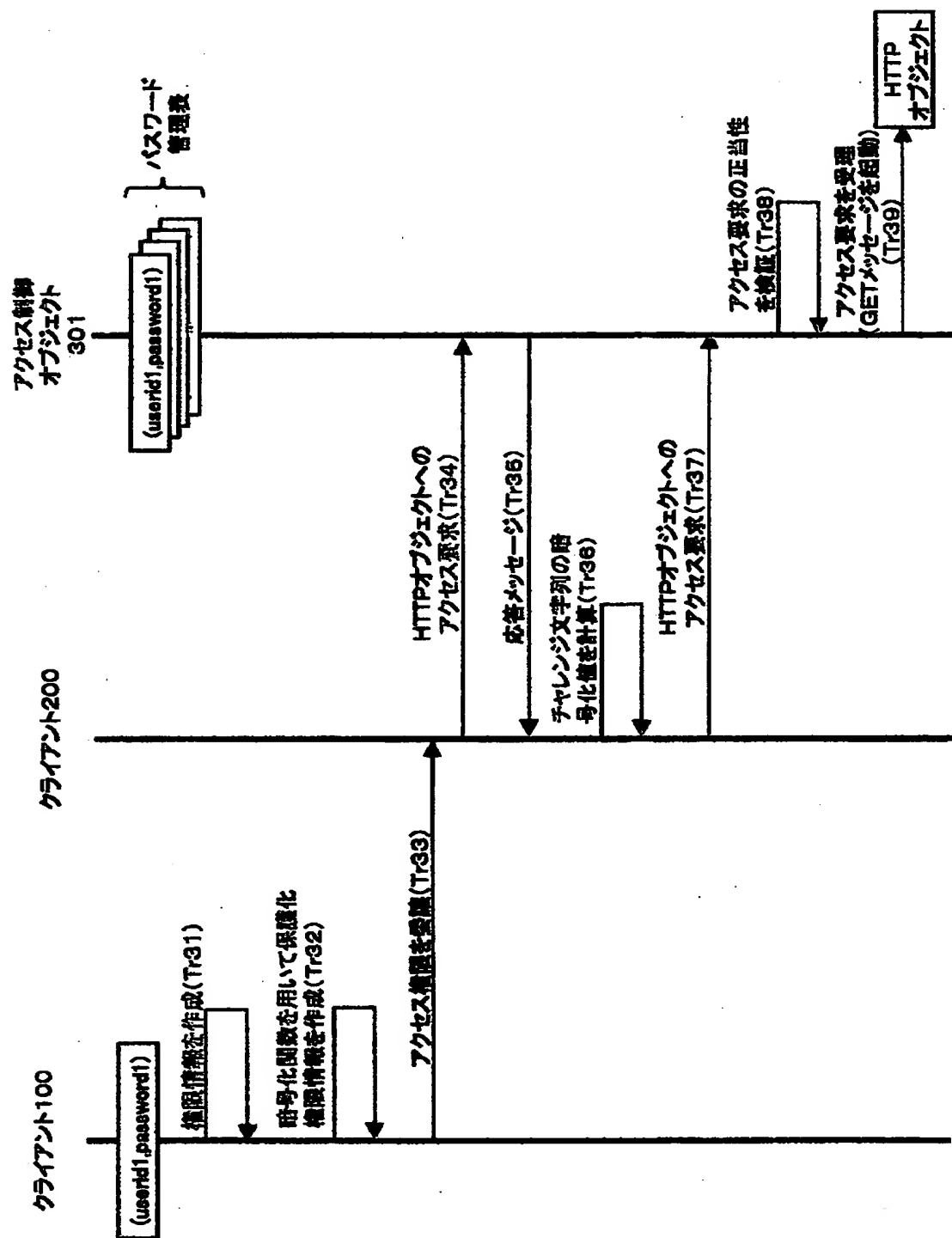
【図 7】



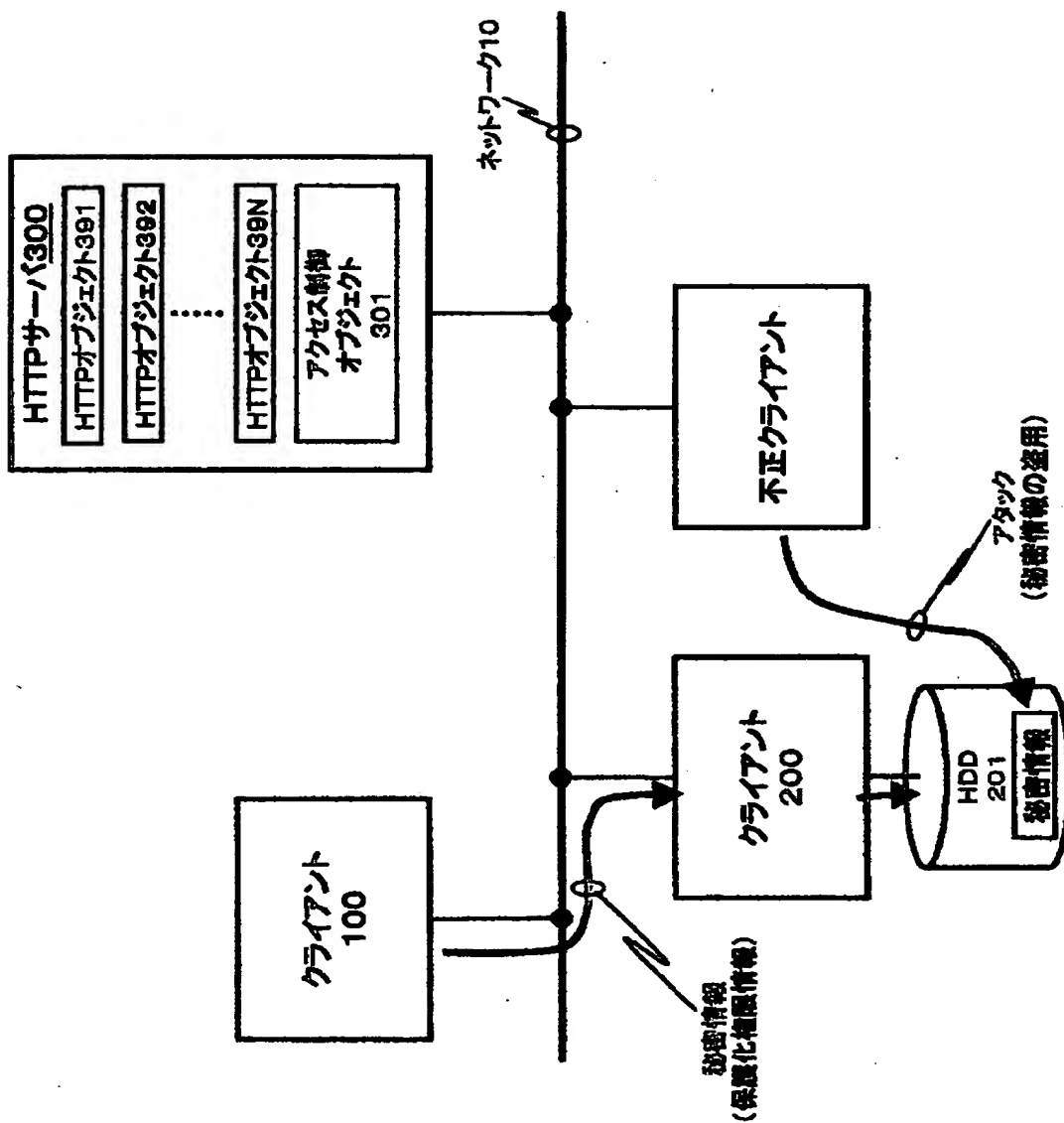
【図 8】



【図 9】



【図 1 0】



【書類名】 要約書

【要約】

【課題】 サービス・オブジェクトが散在するオブジェクト空間上において、オブジェクトに関する権限情報をサブジェクト（ユーザ）間で安全に転送する。

【解決手段】 クライアントとサーバ間では、クライアントのユーザ情報と秘密情報が共有されている。権限情報を委譲するクライアントは、自身が持つ権限内容を弱めた権限情報を生成し、生成した権限情報と秘密情報とを連結したビット列に対して一方向性関数や暗号化関数を適用することで、秘密情報を知らない第三者が改竄不能な保護化権限情報を生成する。保護化権限情報を用いることで、アクセス権限を安全に委譲することができる。また、サーバは、秘密情報を用いて保護化権限情報を解析することで、オブジェクト要求するクライアントが正当か否かを安全に確認することができる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005496]

1. 変更年月日	1996年 5月29日
[変更理由]	住所変更
住 所	東京都港区赤坂二丁目17番22号
氏 名	富士ゼロックス株式会社